BICONTEXTS AND STRUCTURAL INDUCTION

Michael James Livesey

A Thesis Submitted for the Degree of PhD
at the
University of St Andrews

1987

BICONTEXTS AND STRUCTURAL INDUCTION

A thesis submitted to the University

of St. Andrews for the degree of

Doctor of Philosophy

by

Michael James Livesey

Department of Computational Science,

University of St Andrews

September 1986

I, Michael James Livesey, hereby certify that this thesis has been composed by myself, that it is a record of my own work, and that it has not been accepted in partial or total fulfilment of any other degree or professional qualification.

In submitting this thesis to the University of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker.

Signed: Date: 17/9/86

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulation appropriate to the degree of Ph.D.

17/9/86

Signature of supervisor: Date:

Acknowledgements

I would like to thank my supervisor Professor A. J. Cole for his continued support of what must have seemed a lost cause, and I would like to thank my wife Carolyn and my children Tania and Jonathan for their limitless patience in the face of ceaseless adversity.

Abstract

This thesis introduces and explores the notion of *bicontext*, an order-enriched category equipped with a unary endofunctor of order two called *reverse*. The purpose is threefold. First, the important categories that arise in Scott-Strachey denotational semantics have this additional structure, whereby the constructions used to solve "data-type equations" are both limits and colimits simultaneously.

Second, it yields a pleasant "set-theoretic" treatment of algebraic data-types in terms of bicontexts of (1,1) relations rather than pairs of continuous functions. The theory provides a general way of relating bicontexts which serves to connect these particular ones.

Third, the least solutions of data-type equations often have an associated principle of *structural induction*. Properties in such solutions become arrows in the appropriate bicontext, making the defining functor directly applicable to them. In this way the structural induction can be derived systematically from the functor.

Contents

Chapter 1

Introduction

This thesis is an attempt to provide a unifying framework for a number of aspects of the theory of domains, as initiated by Scott in [7] and developed in [8].

Definitions of domain abound in the literature, but the lowest common denominator is an ordered set with a bottom element and the existence of suprema for (some kinds of) directed subsets. Chapter 1 of [1] is an excellent brief survey of domain theory, wherein the author comments upon the proliferation of definitions of domain and points out that it is no more reasonable to expect a single concept of "domain" than of "universe". For this reason we prefer a deliberately vaguer term, and both the title of [8] and current research interests suggest the term *data-type*, or just *type* for short.

The most important technical property of types is the possibility of solving fixed-point equations (more accurately, isomorphisms) over them. The solutions are normally found by some kind of limit construction in a suitable category. All such categories are enriched by additional structure, most notably orderings on the bundles (hom-sets). This was investigated in [13] and pursued in [12]. But there is also a symmetry or bi-directionality about the categories — an arrow from x to y has a *reverse* arrow associated with it from y to x. This gives a somewhat different enrichment; rather than a local enrichment of the individual bundles, we obtain a global enrichment in the form of a unary reversal operator (of course, this operator must behave sensibly with respect to composition and identities). We feel that this structure has not been adequately investigated or exploited to date, although one of the most fundamental categories, namely that of (small) sets and relations between them, possesses it: reverse is just inverse. We call order-enriched categories *contexts* and the symmetric ones *bicontexts.*

One particular effect of the symmtery is the duality between limit and colimit — the limits mentioned above can be viewed as either or both. The authors of [12] try to take account of this, but the result is a somewhat messy concoction of categories and their opposites. Such an outcome is inevitable while the description is being forced into the mould of conventional category theory. Herein, we study categories with added symmetry for their own sake, and although the theory which we develop in Chapter 3 parallels that of ordinary categories, and more especially *Ab*-categories, in many respects, there is significant divergence both at the very fundamental level of the notion of functor and in the notion of limit itself. In fact, it turns out that the natural definition of functor is much weaker than in the conventional case, and that the best way to approach limits is not as a limit — in the "unique existence" sense — at all, although such properties do emerge.

This is the first aspect. The second concerns the subclass of algebraic types. Probably the most important *intuitive* property of a type is the way in which its order structure captures a notion of one value containing a greater or lesser "quantity of information" than another. It is therefore suggestive that the members of a type could actually be sets of "information units", and this idea, elaborated as *information systems* by Scott in [9], leads naturally to consistently-complete ω-algebraic types, hereafter referred to as *simple* types. Such types have many very pleasant properties, but admit only rather weak powerdomain constructions. In [5], Plotkin introduced a wider class of SFP types subsequently shown ([11]) to be the largest subclass of the ω-algebraic types closed under exponentiation.

Of course, it may be said that algebraic types are not fully adequate for denotational semantics because the real numbers, as embodied in Scott's original type ([6]), are not algebraic. But it is possible to construct an ω-algebraic type of reals, starting from the closed intervals with rational endpoints, which we feel in many respects is preferable as a *computational* model because it distinguishes between, for example, 1 and 0.9˙. We describe this type in more detail in Chapter 2.

An algebraic type has a basis consisting of its *compact (isolated, finite)* points, which in fact determines the whole type — we call it the *centre* of the type. The appeal of the information system approach is that much of the behaviour of simple types is describable entirely in terms of their centres. Moreover, the class of orders that can act as centres for simple types is finitely axiomatisable; one of the prices paid for the generalisation to SFP is that the latter's centres are not axiomatisable. In Chapter 5 we study a bicontext of algebraic types, whose arrows are certain (1,1) relations between centres, which have the symmetry that the approximable relations of [9] lack. This turns out to be pleasant in four ways: (a) it, too, allows a treatment in terms of centres, (b) it allows much of the behaviour of the bicontext of pairs of continuous functions, including the standard solution limit constructions mentioned above, to be mimicked in a purely "set-theoretic" manner (this is the burden of Chapter 6 —in part, it extends [14] to SFP types), (c) it gives rise to various notions of effective type, akin (though not identical) to those of [10], but in a seemingly more natural way, and (d) it provides the right backdrop against which to introduce the third aspect, induction.

There is a canonical solution of any type-equation which is generated from the one-point type 0 and has the property that its identity function is the least fixed-point of the functional (functor) embodied in the right-hand side of the equation. This fact may then be used, via fixed-point induction, to justify a principle of *structural induction* on the solution type, often called therefore an *inductive* solution. However, this process is not systematic for two reasons: one is precisely that the functor is generally defined only on pairs of continuous functions on the solution, which bear no obvious relation to *properties*, and the other is that the justification referred to above deals in properties of yet a third kind of object, namely *single* functions over the solution (see [4] for details). An example of this is the class of E_∞ types, defined as the inductive solution of an equation of the form

$$X = A + (X \to X)$$

where A is some constant type independent of the variable type X. What principle of structural induction does it have, and how is it justified? In the conventional setting, the best we can probably do is to observe that E_∞ is also the inductive solution of the equation

$$X = A + (E_\infty \to X)$$

which can be shown to have the induction principle whose step is $(P \subseteq E_\infty)$

$$A \subseteq P \ \& \ \forall f \in (E_\infty \to E_\infty)[(\forall x \in E_\infty,\ fx \in P) \Rightarrow f \in P]$$

which is ad hoc to say the least.

The outstanding advantage of the bicontext of Chapter 5 is that properties on the solution are themselves arrows on the bicontext, rendering such properties valid arguments for the defining functor. This makes the functor generate an induction principle which is already justified by the general theory. Prior to this we take a very general look in Chapter 4 at the phenomenon of induction on a complete order, by relating induction principles to well-orderings and transfinite induction.

Finally, in Chapter 7 we discuss some applications and examples, including E_∞ again.

Chapter 2

Background Material

In this chapter we present all the standard definitions and results that we shall need in the sequel. They are mainly concerned with orders and categories.

2.0 Basics

Miscellaneous Notation.

We declare here various basic items of notation and terminology.

- Sometimes it will be convenient, provided that no confusion results, to use the same lower-case Roman letter for a set and a typical member of it (in the latter case, possibly decorated) — eg: $x_1 \in x$.
- Pow$X$ will denote the powerset of the set X.
- Ord will denote the class of ordinal numbers, with members denoted by lower-case Greek letters.
- In addition to the normal set-inclusion and set-union symbols, $\subseteq, \cup$, we shall also use $\subseteq_{\text{fin}}$ to mean "is a *finite* subset of" and $\uplus$ for disjoint union.
- $f : X \rightharpoonup Y$ means that f is a *partial*-function from X to Y.
- When the meaning is clear, we may sometimes use the quantifiers $\forall, \exists$ in an *adjectival* sense, eg: $x \geq \forall y \in Y$ meaning "x is greater-equal every y in Y. Or $x\ r\ ;\ s\ z \;\Rightarrow\; x\ r\ \exists y\ s\ z$ etc.

Sets and Classes.

We have no wish to become involved in delicate foundational questions herein. Suffice it to say that we assume a background of ZF set theory, and that a class will tacitly be taken to be a set when a particular manipulation requires it. We may adopt the common practice of referring to sets as "small" and proper classes as "large". Thus, for example, in the phrase "the class of spaces", the spaces must necessarily be small, and the class itself large. Also, explicit dependence on the Axiom of Choice arises at a few points, and will be indicated by the annotation "(AoC)".

Tuples.

Tuples will occur frequently. If I, X are classes, an *I-tuple* (in X) is a function $x : I \to X$ where the members of I are thought of as *indexing* the images under x. Thus I is the *index-class* or *arity* of x, and x applied to i will normally be written as x_i rather than xi. When I is a set (which it usually will be), x itself is a set, and the class X^I of all I-tuples in X exists.

If I is a set, and the x_i are sets whose members have some significance, we can form their Cartesian product $\prod x$ or $\prod_{i\in I} x_i$, each element of which is itself an I-tuple in $\bigcup_{i\in I} x_i$. Then, if $\mathcal{E}[i]$ is some expression involving i and denoting, for each $i \in I$, an element of x_i, we shall write $\langle \mathcal{E}[i]\rangle_{i\in I}$, or just $\langle \mathcal{E}[i]\rangle_i$, for the tuple $\lambda i.\mathcal{E}[i] \in \prod x$. In case $I = \{i_1, \ldots, i_n\}$ is finite, we may also write $\langle \mathcal{E}[i_1], \ldots, \mathcal{E}[i_n]\rangle$, and $x_{i_1} \times \cdots \times x_{i_n}$ for the product.

If all the x_i are the same x, the product is a *power*, x^I.

There are certain functions associated with a product. The (component) *projection* functions are $(\cdot)_i : \prod x \to x_i$ for each $i \in I$, and the *superimposition* functions are, for each $J \subseteq I$, $\bar{x} \in \prod x$,

$$\bar{x}[J] : \prod_{j\in J} x_j \to \prod x : x' \mapsto \bar{x}[J/x']$$

where $\bar{x}[J/x'] = \langle \text{if } i \in J \text{ then } x'_i \text{ else } \bar{x}_i \rangle_{i \in I}$ (When $J = \{j\}$, write j instead of J). And if f is an I-tuple of functions such that $f_i : x_i \to y_i$ for each $i \in I$, the *product function* is

$$\prod f : \prod x \to \prod y : \langle x_i \rangle_i \mapsto \langle f_i x_i \rangle_i$$

When I is finite, the x_i can be named explicitly, and therefore the tuple x still exists if the x_i are large, even though X does not exist. The product can also be formed, and the above discussion remains valid. We shall therefore use the tuple terminology uniformly for both cases, with the understanding that if the tuple components are large, I must be finite.

Tuples can be combined. If, for $I \cap J = \emptyset$, x is an I-tuple and y is a J-tuple, define the *catenation* of x, y as the $I \cup J$-tuple

$$x +\!\!+ y = \langle \text{if } k \in I \text{ then } x_k \text{ else } y_k \rangle_{k \in I \cup J}$$

Binary Relations.

Given classes X and Y, a *relation from X to Y* is a subclass of $X \times Y$. We may write $x\ r\ y$ for $\langle x, y \rangle \in r$.

We define the following notation. For $r \subseteq X \times Y,\ s \subseteq Y \times Z,\ X' \subseteq X$:

- $r^{-} = \{\langle y, x \rangle \mid x\ r\ y\}$
- $!r = \{x \in X \mid x\ r\ \exists y \in Y\}$ — the *left-set* of r
- $r! = !(r^{-})$ — the *right-set* of r
- $X'_X = \{\langle x, x \rangle \mid x \in X'\} \subseteq X \times X$
- $r\,;s = \{\langle x, z \rangle \mid x\ r\ \exists y\ s\ z\ \exists y \in Y\}$
- r is *singular* iff $x\ r\ y\ \&\ x\ r\ y' \Rightarrow y = y'$

We shall make the obvious identification between the subsets of X and those of X_X (so X_X is identified with X).

We list without proof some fundamental properties of relations. In the following, r, r' are from X to Y, s is from Y to Z, R, S are classes of relations from X to Y and Y to Z respectively, and $p \subseteq X$, $q \subseteq Y$.

- $p\,;r \cap r\,;q = p\,;r\,;q$. In particular, $p\,;q = p \cap q$
- $!r\,;r = r\,;r! = r$
- $!p = p! = p$
- r is singular iff $r^-\,;r \subseteq Y$
- $(\bigcup R)^- = \bigcup(R^-)$, $\quad (\bigcap R)^- = \bigcap(R^-)$
- $\bigcup R\,;\bigcup S = \bigcup(R\,;S)$
- $\bigcap R\,;\bigcap S \subseteq \bigcap(R\,;S)$ with equality if $\bigcap R$ is singular
- $!r = r\,;r^- \cap X$, so for singular r, $r! = r^-\,;r$
- In case $X = Y$ and r is singular, $X \cap r$ is the set of fixed-points of r
- If r, r' are singular with $r \subseteq r'$ and $!r = !r'$, then $r = r'$

We shall also write sr for $(r\,;s)!$. This is of particular importance in the case of rp, $p \subseteq X$, when it is the set of r-relatives of elements of p.

Polyadic Functions.

First we make some comments about functions of many arguments, *polyadic* functions.

The usual way of treating a polyadic function is as the corresponding monadic function over the appropriate product class. However, these two functions are not equivalent for all purposes, so we shall take the concept of polyadic function as basic (as defined below) and treat monadic functions as a special case. Thus the term "function" shall in general mean "polyadic function".

Let X be an I-tuple of classes, Y a class. A *polyadic function* $f : X \to Y$ is a function $\hat{f}$ from the product of X to Y. By our identification of X with its product, we can think of f as actually being a function over X.

For index i and $a \in X$, we define the *projection* $f_{a,i} : Xi \to Y$ by

$$f_{a,i} x_i = f\, a[i/x_i]$$

A property P of monadic functions can be extended in two ways to polyadic functions:

(1) f has P iff $\hat{f}$ does.

(2) f has P *at* $i \in I$ when $f_{a,i}$ has P for every $a \in X$, and f has P when it has P at every $i \in I$.

If the two extensions are equivalent, we shall that P is *insensitive*. Otherwise we shall distinguish them as *mono-P* for (1) and *poly-P* for (2), although (1) will be the default mode if qualification is absent.

2.1 Orders

An *pre-ordering* on a set X is a binary relation $\leq$ on X which is *reflexive* $(X \subseteq \leq)$ and *transitive* $(\leq ; \leq \subseteq \leq)$. If it is also *anti-symmetric* $(\leq \cap \leq^{-} \subseteq X)$, it is an *ordering*.

We write $x < y$ for $x \leq y$ and $x \neq y$; also $\geq$ $(>)$ is $\leq^{-}$ $(<^{-})$. Notice that the relation $\geq$ is also an ordering. Elements x, y of X are *comparable* when $x \leq y$ or $y \leq x$. We shall call this system of notation for an ordering "pointed notation". An alternative is the following "square notation"

$$\begin{array}{ccc} \sqsubseteq & \longleftrightarrow & \leq \\ \sqsupseteq & \longleftrightarrow & \geq \\ \sqsubsetneq & \longleftrightarrow & < \\ \sqsupsetneq & \longleftrightarrow & > \end{array}$$

Some items of notation introduced below will vary according to the system employed — the square form will appear parenthesised after the pointed form. And of course, either kind of symbol may be decorated in various ways.

An *(pre)order* is a pair $\langle X, \leq \rangle$, with $\leq$ an (pre-)ordering on X. It is a *chain* when any two elements are comparable. $\langle X, \geq \rangle$ is the *dual* order of $\langle X, \leq \rangle$. Any concept associated with an order has a dual, which is that concept applied to the dual order.

2.1.1 PROPOSITION. *If $\langle X, \leq \rangle$ is a preorder, $\sim \stackrel{\text{def}}{=} \leq \cap \geq$ is an equivalence, and $X/{\sim}$ then becomes an order by*

$$x/{\sim} \leq y/{\sim} \Leftrightarrow x \leq y$$

(here $x/{\sim}$ is the $\sim$-class of x).

This notion of order is what is commonly termed a partial-order in the literature. The latter term is clumsy and unnecessary, resulting only from the historical accident that chains were originally called orders. For us, this sub-class is less interesting, so we use the simpler term for the larger class.

Henceforth, we shall only deal with orders unless stated otherwise, and shall refer to the order X when the ordering is understood. Even where two or more orders are mentioned in the same context, we shall use the same symbol where to do so causes no ambiguity.

A fundamental construction on orders is that of product. If X is an I-tuple of orders, its *product* is the product of the sets X_i with the ordering defined by:

$$x \leq y \Leftrightarrow \forall i \in I,\ x_i \leq y_i$$

We shall employ all the notation that we introduced for the product of sets.

We now list some fundamental concepts associated with orders. In the following, X is an order, $x \in X$ and $Y, Y' \subseteq X$

- The *lower shadow* of x, $\langle x]$, is $\{y \in X \mid y \leq x\}$. We shall use $\langle Y]$ to mean $\bigcup_{y \in Y} \langle y]$. The dual is the *upper* shadow, $[x\rangle$.

- x is an *upper bound* of Y iff $Y \subseteq \langle x]$ (write $Y \leq x$ for short, and $Y \leq Y'$ when $Y \leq \forall y' \in Y'$). The dual is *lower* bound.

- $y \in Y$ is *maximal* when $y \not< \forall y' \in Y$. The dual is *minimal* and *extremal* means either maximal or minimal.

- x is *top (greatest)* in x is an upper bound of Y and $x \in Y$. The dual is *bottom (least)*. The bottom of X itself we shall generally call 0.

- x is the *least upper bound (supremum, sup)* of Y iff it is the least of all upper bounds of Y. Write $\bigvee Y$ ($\bigsqcup Y$) for x in this case. If $Y = \{y_1, ..., y_n\}$, we may also write x as $y_1 \vee \cdots \vee y_n$ ($y_1 \sqcup \cdots \sqcup y_n$). The dual is *greatest lower bound (infimum, inf)*, written $\bigwedge$ ($\sqcap$).

- Y is *hereditary* iff $\langle Y] \subseteq Y$. The dual is *persistent*.

- Y is *consistent* iff Y has an upper bound.

- Y is *directed* iff it is non-empty, and any two elements of Y have an upper bound *in* Y.

- Y is an *ideal* when it is either empty or hereditary and directed.

- Y *dominates* Y' iff $Y' \subseteq \langle Y]$. When Y dominates $\{x\}$, we say simply that it dominates x, and vice versa.

- Y *entails* Y' ($Y \vdash Y'$) iff every upper bound of Y is $\geq \exists y \in Y'$.

- Y' is a *roof* of Y iff $Y' \geq Y$, $Y \vdash Y'$, and Y' is finite.

- Y is $\mathcal{U}$*-closed* iff every *finite* non-empty $Y' \subseteq Y$ has a roof inside Y.

- Y is *projective* iff every $\langle x] \cap Y$ ($x \in X$) is directed or empty.

It will be convenient to introduce some systematic terminology, as follows:

If $SS[X]$ is some generic family of subsets of order X, and $Y \subseteq X$, we say that Y is SS*-closed* to mean that whenever $Y' \in SS[Y]$ has a sup in X, it is in Y. In a similar way, we say that X itself is SS*-complete* to mean that every $Y \in SS[X]$ has a sup. In particular, monotonic families ($Y \subseteq X \Rightarrow SS[Y] \subseteq SS[X]$) enjoy the following properties:

2.1.2 PROPOSITION. *For monotonic SS*

(1) *If X is SS-complete and $Y \subseteq X$ is SS-closed, Y is itself SS-complete*

(2) *The collection of SS-closed subsets of X is a closure system, whose closure operation we shall write as $SS\langle ... \rangle$*

PROOF:

(1) Let $Y' \in SS[Y]$. Then $Y' \in SS[X]$, so Y' has a sup, say y', in X. By the closure, $y' \in Y$. But then y' is obviously the Y-sup of Y'.

(2) Let YY be a family of SS-closed subsets of X, with intersection Y. Let $V \in SS[Y]$ have sup v. Then $V \in SS[Y']$, $\forall Y' \in YY$, whence $v \in Y'$. It follows that $v \in Y$. ∎

The following special cases are of importance:

- SS is all subsets. In this case, we simply omit it.
- SS is all directed subsets. We refer to it as "D".
- SS is all non-empty consistent subsets. We refer to it as "almost".
- SS is all subsets of size 2. We refer to it as "dual".
- SS is all consistent subsets of size 2. We refer to it as "dualmost".
- SS is the empty set. We call this case "0"

Note that almost-closure requires only consistent subsets *of* Y to have their X-sup (if it exists) within Y. To say that X is 0-complete is to say that 0 exists; to say that $Y \subseteq X$ is 0-closed is to say that if 0 exists in X, it is in Y. All of these families are monotonic.

We can also evidently characterise the dual-closure of $Y \subseteq X$ as the set $\{\bigvee Y' \mid Y' \subseteq_{\mathrm{fin}} Y\}$, which has the same set of upper bounds as Y.

We list without proof some obvious facts about these concepts. Again, X is an order with typical element x and subset Y.

- $\emptyset$ is consistent provided X is non-empty, with every x as an upper bound

- 0, if it exists, is $\bigvee \emptyset$
- $\{x\}$ is consistent, with sup x
- If Y has a top, it is also its sup.
- Y is directed iff every finite subset of it has an upper bound in Y
- If $Y \subseteq Y'$, then $\langle Y] \subseteq \langle Y']$.
- $Y \subseteq \langle Y]$.
- $\langle Y]$ is hereditary.
- If YY is a set of subsets of X, $\langle \bigcup YY] = \bigcup_{Y \in YY} \langle Y]$.
- If Y dominates Y', then it dominates $\langle Y']$
- Y dominates Y' iff Y dominates every point of Y'
- If directed D dominates finite Y, it contains an upper bound of Y
- If Y' is a roof of Y, and Y has a sup, then $\bigvee Y \in Y'$
- If Y, Y' have equal sups, they have common upper bounds. Conversely, if they have common upper bounds and either has a sup, the other has the same sup.

2.1.3 PROPOSITION.

(1) *If Y has a roof, it has a smallest roof, $\mathcal{U}Y$*

(2) *The $\mathcal{U}$-closed sets are a partial closure system. We write $\mathcal{U}^*Y$ for the $\mathcal{U}$-closure of Y, when it exists*

(3) *A directed union of $\mathcal{U}$-closed sets is $\mathcal{U}$-closed*

(4) *Any finite order is $\mathcal{U}$-closed*

PROOF:

(1) Let R, S be minimal roofs of Y, with $r \in R$. Then $r \geq Y$, so $r \geq \exists s \in S$, and in turn $s \geq \exists r' \in R$. But then every upper bound of $R \geq r$ is also $\geq r'$, so $R \setminus \{r\}$ is a roof of Y, which is impossible unless $r = r'$, whence $r = s$. Thus $R \subseteq S$; conversely, $S \subseteq R$.

(2) By a *partial* closure system we mean closed under arbitrary *non-empty* intersections, but X itself may not be closed. Thus Y has a closure iff some closed

$Y' \supseteq Y$. Let $YY (\neq \emptyset)$ be a set of $\mathcal{U}$-closed subsets of X with intersection Y. Let $A \subseteq_{\text{fin}} Y$. Then A has a roof inside every Y' in YY, so it has a smallest roof in every such Y', hence in Y.

(3) Let YY be a directed set of subsets of X with union Y. Then if $A \subseteq_{\text{fin}} Y$, $A \subseteq \exists Y' \in YY$, so has a roof in Y', hence in Y.

(4) If $Y \subseteq X$, the set of all upper bounds of Y is a roof for it. ∎

Of course, not every consistent subset of an order need have a sup (see, for example, Diagram 2.2).

2.1.4 DEFINITION.

(1) *A* (data-)type *is an order which is both D- and* 0*-complete.*

(2) *If X is a type and $Y \subseteq X$, Y is a* subtype *when it is both D- and* 0*-closed.*

This concept is usually referred to as a "complete partial-order" in the literature. The term is a misnomer, because "complete" suggests that every subset has a sup (and should , in fact, be synonymous with "complete lattice").

Examples.

(1) Given any set X, we can construct from the *flat* type $X_\perp$ by adjoining an extra bottom element to X — see Diagram 2.1.

(2) Let $\omega + 1$ have its natural order, and adjoin extra elements $1', 2'$ with $\{1, 1'\} \leq \{2, 2'\}$, $0 \leq 1'$ and $2' \leq \omega$ — see Diagram 2.2.

(3) For any set X, its powerset, Pow$X$, is a type under the inclusion ordering. It is complete with bottom $\emptyset$ and top X; its sup and inf operations are union and intersection respectively.

(4) If X is an order, its $\mathcal{U}$-closed subsets together with $\emptyset$ are a subtype of Pow$X$.

The following notation will prove useful. Given a type X and $Y \subseteq X$, we shall write $D \nearrow_Y d$ to mean that D is a directed subset of Y with sup d (not necessarily in Y). When $Y = X$, we shall write simply $D \nearrow d$.

2.1.5 LEMMA. *If X is a type and $Y \subseteq X$ is such that every finite subset of Y has a sup, then Y has a sup.*

PROOF: Let $YY = \mathrm{dual}(Y)$. Then YY is directed because the subset

$$\{\bigvee Y_1, ..., \bigvee Y_n\} \subseteq YY$$

has upper bound $\bigvee(Y_1 \cup \cdots \cup Y_n)$. So YY has a sup, and since Y and YY have the same upper bounds, Y has the same element as sup. ∎

2.1.6 COROLLARY. *Type X is almost-complete iff it is dualmost-complete.*

PROOF: The forward implication is trivial. To establish the reverse, we first show that every consistent finite subset of X has a sup, by induction on the size of the subset. Then if $Y \subseteq X$ is consistent, every finite subset of Y is consistent, so has a sup. Now use 2.1.5. ∎

2.1.7 LEMMA. *If X is an almost-complete type and $Y \subseteq X$ is non-empty, then $\bigwedge Y$ exists.*

PROOF: Let Y' be the set of all lower bounds of Y. Because $Y \neq \emptyset$, Y' is consistent, and so has a sup. It is easily checked that $\bigvee Y' = \bigwedge Y$. ∎

We define some more concepts associated with types. In the following, X is a type, with typical element x, subset Y and directed subset D:

- D *passes* x iff $x \leq \bigvee D$
- Y is *pre-compact* iff for any $y \in Y$ and any D which passes y, there is directed $D' \subseteq Y$ with sup y and dominated by D. Y is *compact* iff it is finite and pre-compact. If $\{x\}$ is compact, we say that x is compact. Notice that 0 is compact.
- The *support* of Y is $Y| = \{\text{compact } a \in (Y|\}$. We write $x|Y'$ for $\{x\}|Y'$, and $Y|Y'$ for $Y| \cap Y'$.

- The *centre* of X, X°, is the set of all its compact points, except 0.
- Y is *algebraic* iff every $y \in Y$ has $y|Y \nearrow y$. Of special importance is the case where X itself is algebraic.

The compact points are sometimes called the *isolated* points (sometimes also the "finite" points, but this is another very over-worked term). Non-compact points are often called *limit* points.

We have the following properties of supports:

- $X| = X^\circ \cup \{0\}$
- $x \in x|$ iff x is compact
- If $x \leq x'$ then $x| \subseteq x'|$

2.1.8 PROPOSITION.

(1) *Let $\bigvee Y = y$; then $Y| \subseteq y|$, with equality if Y is directed*

(2) *If $Y \subseteq X^\circ$ and, for every non-0 $x \in X, x|Y \nearrow x$, then $Y = X^\circ$ and X is algebraic.*

PROOF:

(1) The fact that $\langle Y] \subseteq \langle y]$ gives the inclusion. Now if Y is directed and $a \in y|$, Y passes a and so, because a is compact, Y dominates a, ie: $a \in Y|$.

(2) Let $a \in X^\circ$. Then $a = \bigvee(a|Y)$, which implies that $a \leq \exists a' \in a|Y$, which implies in turn that $a = a'$. Thus $a \in Y$. Hence $Y = X^\circ$, and the result follows. ∎

In general, a point in a type may be the sup of its support without the support being directed. For consider the type X comprising two copies of $\omega + 1$, with 0 common, ordered as shown in Diagram 2.3; ω' is such a point.

2.1.9 PROPOSITION.

(1) *A union of pre-compact sets is pre-compact. In particular, any set of compact points is pre-compact.*

(2) *A finite set is compact iff each of its points is compact.*

(3) *The sup of a compact set is compact.*

In an algebraic type we also have:

(4) *$\mathcal{U}$ of a compact set is compact.*

(5) *The $\mathcal{U}$-closure of any set of compact points consists of compact points.*

PROOF:

(1) is trivial.

(2) We need only prove the "only if". Let X be a type with compact $Y \subseteq X$, and let $y \in Y$ be passed by directed $D \subseteq X$. Then there is directed $D' \subseteq Y$, with sup y, dominated by D. Since D' is finite and directed, $y \in D'$. Thus y is dominated by D, whence y is compact.

(3) Let $yy = \bigvee Y$, Y compact. Let yy be passed by directed D. Then each $y \in Y$ is passed by D, therefore dominated by D. So D contains an upper bound of Y, whence yy is dominated by D.

(4) Suppose $A \subseteq X$ is compact, and $y \in \mathcal{U}A \setminus X^\circ$. Let $x \geq y \geq A$. Then $y|$ dominates A, hence contains some $a \geq A$. Therefore $\exists b \in \mathcal{U}A$ with $a \geq b \geq A$. Thus $x \geq y > a \geq b \geq A$, so $\mathcal{U}A \setminus \{y\}$ is a roof of A, contradiction.

(5) If $Y \subseteq X|$, $\mathcal{U}^*Y \cap X|$ is $\mathcal{U}$-closed, by (2). ∎

In connection with part (3), [11] gives the explicit construction of $\mathcal{U}^*Y$ as $\bigcup\limits_n \mathcal{U}_n Y$ where

$$\mathcal{U}_0 Y = Y$$

$$\mathcal{U}_{n+1} Y = \bigcup\{\mathcal{U}A \mid A \subseteq_{\text{fin}} \mathcal{U}_n Y\}$$

Furthermore, it is obvious that $\mathcal{U}^*Y \cap [Y\rangle$ is $\mathcal{U}$-closed, whence $\mathcal{U}^*Y \subseteq [Y\rangle$.

2.1.10 DEFINITION. *A type is:*

(1) ω-algebraic *iff it is algebraic and has a countable centre.*

(2) *SFP iff it is ω-algebraic and every compact set has a finite $\mathcal{U}$-closure (and therefore a fortiori a roof).*

(3) simple *iff it is almost-complete ω-algebraic*

2.1.11 PROPOSITION. *Simple implies SFP*

PROOF: Let X be simple with compact subset Y. Let $YY = \mathrm{dual}\langle Y\rangle$. Then $Y \subseteq YY$, which is finite. If $Y' = \{\bigvee Y_1, ..., \bigvee Y_n\} \subseteq YY$ (each $Y_i \subseteq_{\mathrm{fin}} Y$), then:

(i) Y' inconsistent: $\emptyset$ is a roof of Y' within YY. (ii): Y' consistent: $\{\bigvee Y'\}$ is a roof of Y' within YY because $\bigvee Y' = \bigvee \bigcup_{i=1}^{n} Y_i$.

So YY is $\mathcal{U}$-closed. ∎

Notice that $\mathrm{dual}\langle Y\rangle$ is just $\mathcal{U}_0 Y$ above, and is already $\mathcal{U}^* Y$.

We now prove an important property of algebraic types.

2.1.12 THEOREM. *An hereditary subset of an algebraic type is pre-compact*

PROOF: Let A be algebraic with $X \subseteq A$, and let $D \nearrow dd \geq x \in X$. Then $x| \nearrow_X x$, and every $a \in x|$ is passed, and therefore dominated, by D. Hence $x|$ is dominated by D. ∎

2.1.13 COROLLARY. *An algebraic type is pre-compact itself.*

2.2 FUNCTIONS OVER ORDERS

In this section we look at relations and functions between orders.

Let X, Y be orders (typical elements x, y), and let $r \subseteq X \times Y$ a relation. Then r is:

- *monotonic* iff $x \leq x'$ & $x\ r\ y$ & $x'\ r\ y' \Rightarrow y \leq y'$.
- *bimonotonic* iff r and r^- are both monotonic.
- an *isomorphism* iff it is bimonotonic, total and onto. We may then write $r : X \cong Y$. Moreover, $X \cong Y$ will mean $\exists r : X \cong Y$, in which case X, Y are said to be *isomorphic*.

When r is a function, it is:

- *projective* when every non-empty $\{x \mid rx \leq y\}$ has a top.
- *central* on $X' \subseteq X$ when X, Y are algebraic and $r(X'^{\circ}) \subseteq r(X')^{\circ}$.

We have the following basic properties:

2.2.1 PROPOSITION. *Let function $f : X \to Y$ be monotonic. Then:*

(1) *A monotonic relation is singular, and a bimonotonic one is (1,1). Thus a bimonotonic function is an injection (ie: one-one).*

(2) *An isomorphism is a bijection, whence its inverse is also an isomorphism.*

(3) *If $S \subseteq X$ with $\bigvee S = x$, and $\bigvee f(S) = y$, then $y \leq fx$.*

(4) *If $D \subseteq X$ is directed, then $f(D)$ is directed.*

PROOF:

(1) $x\ r\ y\ \&\ x\ r\ y' \Rightarrow y \leq y' \leq y$.

(2) Immediate from (1).

(3) Every $fs \leq fx (s \in S)$ by monotonicity, so fx is an upper bound of $f(S)$.

(4) Let $fd', fd'' \in f(D)$, for $d', d'' \in D$. Then $d', d'' \leq \exists d \in D$, so by monotonicity of f, $\{fd', fd''\} \leq fd$. ∎

Now also let T be an I-tuple of types, and let $D \nearrow_X x$, $S \subseteq X$ with sup s, and let $f : X \xrightarrow{!} Y$, $g : T \dashrightarrow Y$.

- f is *strict* when $f0 = 0$.
- f is *continuous* (is a *map*) when it is monotonic and $fd = \bigvee f(D)$.
- f is *linear* when it is monotonic and $fs = \bigvee f(S)$.
- g is *definite* when it is monotonic and

$$gt = 0 \quad \Rightarrow \quad t_i = 0 \quad \exists i \in I$$

$$0 < gt \leq gt' \quad \Rightarrow \quad t \leq t'$$

Thus a linear function preserves all sups which exist, not just directed ones. In particular, it is strict. We also have

2.2.2 PROPOSITION. *For (monadic) monotonic $f : X \to Y$:*

(1) *If it is projective, it is linear.*

(2) *If it is linear and X is complete, it is projective.*

PROOF:

(1) Let $S \dot{\subseteq} X$ with sup s, and let $y \geq f(S)$. Let x be the top of $\{x' \mid fx' \leq y\}$ ($\supseteq S$). Then $s \leq x$, whence $f(S) \leq fs \leq fx \leq y$, giving $fs = \bigvee f(S)$.

(2) Given $y \in Y$, let $x = \bigvee\{x' \mid fx' \leq y\}$. Then $fx = \bigvee\{fx' \mid fx' \leq y\} \leq y$. Thus x is top in $\{x' \mid fx' \leq y\}$.

∎

Notice that a monadic map over an algebraic type is determined by its values on the compact points, because every $x = \bigvee(x|)$. In fact, it is determined by its *graph*, where

$$\text{graph}(f : X \to Y) = \{\langle a, b\rangle \in X| \times Y| \mid b \leq fa\}$$

2.2.3 PROPOSITION.

(1) *With the same notation, let f be a bimonotonic map, and let $D \nearrow_{f(X)} d$. Then $d \in f(X)$, with $f^- d = \bigvee f^-(D)$.*

(2) *An isomorphism between types is continuous (and therefore its inverse also)*

PROOF:

(1) Clearly, as in the proof of 2.2.1(4), $f^- D$ is directed, so its sup exists, say d'. But by continuity, $fd' = \bigvee f(f^- D) = \bigvee D = d$.

(2) Now let f be an isomorphism. Let $D \nearrow_X d$. Let $d' = \bigvee fD$. Then by 2.2.1(3), $d' \leq fd$ and $d \leq f^- d'$. But together these imply $fd = d'$. ∎

It is a standard result that continuity is insensitive. This is not the case for strictness, definiteness and linearity, although we do have that polystrict implies monostrict and that a polystrict monolinear function is polylinear. Moreover, a definite monadic function is bimonotonic.

Prime examples of maps are the identity function on any type any constant function, the composite of two maps, and any product of maps.

There is a standard point-wise ordering imposed on the functions between two orders, viz:

$$f \leq g \Leftrightarrow fx \leq gx \quad \forall x \in X$$

2.2.4 DEFINITION. *For types X, Y, the order $(X \to Y)$ is the set of all* maps *from X to Y under the point-wise ordering.*

Now consider the statement template:

For .... types X, Y, $(X \to Y)$ is also a .... type.

It is well known ([2,11]) that the blank can be omitted or filled variously with "almost-complete", "complete", "SFP" or "simple", but *not* with "ω-algebraic".

Now if $F \subseteq (X \to Y)$ is such that for any $x \in X$, $Fx \stackrel{\text{def}}{=} \{fx \mid f \in F\}$ has a sup s_x, then the pointwise sup function

$$s : X \to Y : x \mapsto s_x$$

is continuous and the sup of F.

It follows that directed sups, and in the almost-complete case all sups, in $(X \to Y)$ are the pointwise sups. However, this property does not extend to roofs in the SFP case (ie: $f \in \mathcal{U}F \not\Rightarrow \forall x \in X,\ fx \in \mathcal{U}(Fx)$) as the following example shows.

Let $X = \{0, 1\}$ under the natural ordering and Y be the order in Diagram 2.4. Define $f, g, h : X \to Y$ by

$$f = \{0 \mapsto 0, 1 \mapsto 1\}$$

$$g = \{0 \mapsto 0', 1 \mapsto 1'\}$$

$$h = \{0 \mapsto a, 1 \mapsto b\}$$

Then $h \in \mathcal{U}\{f, g\}$, but $h1 = b \notin \mathcal{U}\{f1, g1\} = \mathcal{U}\{1, 1\} = A$.

A crucial property of endomorphic maps is:

2.2.5 PROPOSITION. *Let X be a type, and let $f \in X \to X$. Then f has a least fixed-point (LFP),* $\operatorname{fix} f$, *given by*

$$\operatorname{fix} f = \bigvee\{f^n 0 \mid n \geq 0\}.$$

If $x \in X$ with $x \leq fx$, f cuts down to the subtype $[x\rangle$. We shall write $\operatorname{fix}_x f$ for the LFP of this restriction of f. Such an x we shall call a *seed* of f.

The LFP is itself a continuous operation, in the sense that $\operatorname{fix}_x f$ is continuous in both x and f.

2.3 GENERATING TYPES

Given any order, we can generate from it an algebraic type with the original order as its centre. If the generator is dualmost-complete, the generated type will be almost-complete, whence if the former is also countable, the latter will be simple. The last case is that used in [9] to allow a simple type to be presented as an information space. The construction is detailed in the next definition.

2.3.1 DEFINITION. *Let X be an order. Define the* type generated *by X to be the set* $\operatorname{Type}(X)$ *all ideals in X under the inclusion ordering.*

2.3.2 THEOREM. $\operatorname{Type}(X)$ *is an algebraic subtype of* $\operatorname{Pow}X$, *with centre* $\{\langle x] \mid x \in X\}$ *(which is therefore isomorphic to X).*

PROOF:

(1) $\operatorname{Type}(X)$ is a subtype of $\operatorname{Pow}X$: Clearly $\emptyset$ constitutes 0. Now let D be a directed (with respect to $\subseteq$) subset of $\operatorname{Type}(X)$. If $x', x'' \in \bigcup D$, then they are both in some $d \in D$ Thus there is an $x \in d$ with $x', x'' \leq x$. Since $x \in \bigcup D$,

$\bigcup D$ is directed. Furthermore, if $x \in \bigcup D$, $x \in \exists d \in D$, so $\langle x] \subseteq d \subseteq \bigcup D$, ie: $\bigcup D$ is hereditary.

(2) Type(X) is algebraic with centre $X' = \{\langle x] \mid x \in X\}$: Every $\langle x]$ is obviously in Type(X), and the bijection $x \longleftrightarrow \langle x]$ is clearly an isomorphism between X and X'. Now let D be a directed subset of Type(X) which passes $\langle x]$. Then $x \in \bigcup D$, whence $x \in \exists d \in D$, whence $\langle x] \subseteq d$, so that D dominates $\langle x]$. Thus $\langle x]$ is compact, ie: $X' \subseteq \text{Type}(X)^{\circ}$. Furthermore, if $s \in \text{Type}(X)$, $s|X' = \{\langle x] \mid x \in s\}$, which is directed because s is, and obviously has union s. The result follows. ∎

2.3.3 THEOREM. *If X is dualmost-complete,* Type(X) *is almost-complete.*

PROOF: Let $S \subseteq \text{Type}(X)$ be consistent, with $S \leq ss$. We claim that the set s defined as

$$\{x \in X \mid x \leq \bigvee s',\ \exists s' \subseteq_{\text{fin}} \bigcup S\}$$

is the sup (within Type(X)) of S. Obviously s is hereditary. Let $x', x'' \in s$, with say

$$x' \leq \bigvee s', \quad x'' \leq \bigvee s''$$

Then $s', s'' \subseteq ss$, so $s' \cup s'' \subseteq ss$. Therefore $s' \cup s''$ has an upper bound in ss (because ss is directed), whence (by dualmost-completeness) it has a sup, which is clearly in s and an upper bound of $\{x', x''\}$. Thus s is directed, ie: $s \in \text{Type}(X)$. Finally, if $x \in s$ with $x \leq \bigvee s'$, then $s' \subseteq ss$, so has an upper bound in ss (as above). Because ss is hereditary, $x \in ss$, ie: $s \subseteq ss$. Thus s is indeed the sup of S. ∎

2.3.4 COROLLARY. *When X is countable dualmost-complete,* Type(X) *is simple.*

Example.

Let $\mathbb{R}$ be the set $\{[r, s] \mid r \leq s \in \mathbb{Q}\}$, where $\mathbb{Q}$ is the set of rational numbers together with $\pm\infty$, under the superset ordering $\supseteq$. Obviously $\mathbb{R}$ is dualmost-complete.

As we mentioned in Chapter 1, Type($\mathbb{R}$) is an algebraic version of the Scott type of real numbers. The difference is that every $[r, s] \in \mathbb{R}$ has four distinct representatives in Type($\mathbb{R}$), viz. $\{[r', s'] \mid r' \leq r, s \leq s'\}$, $\{[r', s'] \mid r' < r, s \leq s'\}$, $\{[r', s'] \mid r' \leq r, s < s'\}$ and $\{[r', s'] \mid r' < r, s < s'\}$ which we shall write as $[r, s], (r, s], [r, s)$ and (r, s) respectively. The ordering amongst these is $(r, s) \leq \{(r, s], [r, s)\} \leq [r\ s]$ with the two middle ones incomparable. In the case of a single rational number (ie: $r = s$) the four representatives correspond to: computing the number exactly, computing a definite upper bound, computing a definite lower bound, and finding no bound but only tending to the number in the limit from either side — the number itself may be the same in each case, but computationally they are very different.

2.4 Categories and Functors

Finally in this chapter, we give the basic definitions of the concepts of category and functor.

A *category*, C, is a class (the carrier, traditionally called the *arrows* of C and written $|C|$ when it needs distinguished from C itself), together with:

- A subclass obj$C$ of C, the *objects* of C ($f, g, \ldots$ are typical arrows, and $x, y, \ldots$ typical objects),
- A ternary relation on $C \times (\text{obj}C)^2$, written $f : x \to y$, and a partial binary operation, called *composition*, on C, written $f \,;\, g$ satisfying:
- $f : x \to y \;\&\; f : x' \to y' \;\Rightarrow\; x = x' \;\&\; y = y'$. x is the *source* and y the *target* of f.
- If $f : x \to y$ and $g : x' \to y'$, $f;g$ is defined iff $y = x'$, in which case $f;g : x \to y'$.
- If $f : x \to y$, $g : y \to z$ and $h : z \to w$ then $(f \,;\, g) \,;\, h = f \,;\, (g \,;\, h)$. We write simply $f \,;\, g \,;\, h$.
- $x : x \to x$ and is an *identity* for composition, ie: $f \,;\, x = f$, $x \,;\, g = g$ for any f with target x or g with source x. Clearly x is the only such identity. We may

refer to an object simply as "1" when it is clear from the context which object is in question, or in order to talk about an arbitrary object.

- If arrows f and g have the same source and target, they are *parallel* $(f||g)$.
- The class $\{f \in C \mid f : x \to y\}$, denoted by $C[x, y]$, is called a *bundle*.
- For objects x, y, z, composition restricts to a binary operation from $C[x, y] \times C[y, z]$ to $C[x, z]$. These are the *local compositions*.

The bundle $C[x, x]$ is the class of *loops* on x. It is a monoid.

If C is a category and $C' \subseteq C$, C' is a *sub-category* of C when:

- $f : x \to y \ \& \ f \in C' \ \Rightarrow \ x, y \in C'$
- If $f, g \in C'$ with $f \,;\, g$ defined, then $f \,;\, g \in C'$

It is *full* when $x, y \in C'$ and $f : x \to y$ implies $f \in C'$, and it is *spanning* when it contains every object. We shall equivocate between a full sub-category and its object class when there is no confusion.

One can form another category from C by turning all the arrows round. The *opposite* category, C_{op}, of C has the same arrows and the same objects, but $f : x \xrightarrow{op} y$ iff $f : y \to x$ and composition $f \,;_{op}\, g = g \,;\, f$.

Later, we shall make use of a particular kind of category, which we now define.

2.4.1 DEFINITION. *A net is a category such that:*

(1) *Given objects x', x'', there are arrows $f' : x' \to x$, $f'' : x'' \to x$ for some x.*

(2) *Given arrows $f' : x \to x'$, $f'' : x \to x''$, there are arrows $g' : x' \to y$, $g'' : x'' \to y$ with $f' \,;\, g' = f'' \,;\, g''$. We say that $\langle g', g'' \rangle$ unifies $\langle f', f'' \rangle$.*

We can also form the product of a tuple of categories — if C is an I-tuple of categories its *product* $\prod C$ has:

- $|\prod C| = \prod_i |C_i|$
- $\mathrm{obj} \prod C = \prod_i \mathrm{obj} C_i$
- $f : x \to y \ \Leftrightarrow \ f_i : x_i \to y_i \ \forall i \in I$

- $f \,;_\Pi\, g = \langle f_i \,;\, g_i \rangle_i$

Again we use all the standard set-product notation.

We now define the notion of *functor*, which is a function between categories preserving the category structure.

Let C, C' be categories and $F : C \to C'$ a function. F is a *pre-functor* when it satisfies:

- Every $F1$ is an object
- $f : x \to y \;\Rightarrow\; Ff : Fx \to Fy$

Thus F maps each $C[x, y]$ to $C'[Fx, Fy]$. We write $F[x, y]$ for the restriction of F to $C[x, y]$, and call these the bundles of F. Then F is a *functor* when it also satisfies (for any composable arrows f, g):

- $F(f \,;\, g) = Ff \,;\, Fg$

Furthermore, F is an *isomorphism* when it is a bijection between $|C|$ and $|C'|$.

When we come to add order structure to categories, the pre-functors will play as important a role as the functors.

It is immediate that the functional composite of (pre-)functors is also one. We shall normally write this composite as juxtaposition, with GF meaning $\lambda f.G(Ff)$.

In line with our conventions about polyadic functions, a polyadic function over a tuple of categories will be a functor precisely when it is a mono-functor. In this case, however, $F[x, y]$ will be thought of as a polyadic function.

Two kinds of functor have special names. A functor from C to itself is called *covariant on* C, and one from C_{op} to C is called *contravariant on* C

We conclude with a definition that will be useful in the sequel.

2.4.2 DEFINITION.

(1) *A bijection between* obj$C$ *and* objC' *is a* coincidence; *C and C' are then* coincident.

(2) *If C, C' are coincident, (pre-)functor $F : C \to C$ is* static *when it extends the relevant coincidence.*

Chapter 3

Contexts and Bicontexts

3.1 Basic Definitions

As we pointed out in Chapter 1, all the relevant categories used in denotational semantics and the associated theory of computation are endowed with orderings on the bundles. Many also have some natural way of "reversing" arrows. In this chapter we pursue both these ideas.

A *context* is a category, C, in which every bundle is a type with respect to which each local composition is continuous and strict. In an obvious way, C_{op} is also a context.

Two immediate consequences of this are that no bundle is empty — there is always the zero arrow $0 : x \to y$ for any objects $x, y \in C$ — and that every bundle is a set (even though the context itself may be large).

An *isomorphism* between two contexts is an isomorphism of the underlying categories with every bundle bimonotonic.

Given context C and $K \subseteq C$, we say K is a *pre-subcontext*, or just *pre-sub* of C iff it is a sub-category of the underlying category of C and every bundle of K is D-closed relative to C. We write $K < C$, and $K[x, y]$ for $K \cap C[x, y]$. If every K-bundle is a subtype of the corresponding C-bundle (ie: $K[x, y]$ contains $0 : x \to y$), then K is a *subcontext (sub)*, and is then a context itself. In this case, we write $K \triangleleft C$.

A *symmetric* context, or *bi*context, is a context equipped with a static contravariant functor $(\cdot)^-$ of the underlying category which is monotonic on each bundle, and of order 2 (ie: every $f^{--} = f$). f^- is the *reverse* of f.

$C-$ will denote the underlying context of bicontext C. If K is a (pre-)sub of $C-$, we say that K is a (pre-)sub of C, and write simply $K \lhd C$ $(K < C)$.

If K is a (pre-)sub of $C-$ and is closed under reversal, it is a *symmetric* (pre-)sub, or (pre-)*bi*sub, of C, and we write $K \lhd^= C$ $(K <^= C)$. In case $K \lhd^= C$, K itself becomes a bicontext. With the obvious meaning, we can refer to full and spanning (pre-)subs (clearly a full pre-sub must be a sub).

We may also apply the term symmetric to functions — if $F : |C| \to |C'|$ is any function, it is symmetric if every $F(f^-) = (Ff)^-$ (equivalently $F(f^-) \leq (Ff)^-$). In such a case we may unambiguously write Ff^-.

An *isomorphism* between bicontexts is a symmetric isomorphism between the underlying contexts.

3.1.1 Proposition. *For any* $x, y \in \mathrm{obj}C$, $(\cdot)^- : C[x,y] \cong C[y,x]$

Examples.

(1) $\mathrm{obj}C <^= C$

(2) $\mathrm{obj}C$ plus all the zeroes $(\mathrm{obj}_0 C) \lhd^= C$

(3) If $X \subseteq \mathrm{obj}C$, $C|X$ will denote the full bisub determined by X, ie: all arrows between objects in X.

There is a simple way in which we can create a bicontext from a context. Given context C, we define the bicontext $\mathrm{Pair}(C)$ as follows.

Its objects are the the same as those of C. Its arrows are pairs $\langle f, f' \rangle$ such that $f : x \to y$, $f' : y \to x$ in C, with $\langle f, f' \rangle : x \to y$. Composition is defined by

$$\langle f, f' \rangle \,;\, \langle g, g' \rangle = \langle f \,;\, g,\ g' \,;\, f' \rangle,$$

reversal by

$$\langle f, f'\rangle^{-} = \langle f', f\rangle$$

and the bundle type by

$$\mathrm{Pair}(C)[x, y] = C[x, y] \times C[y, x].$$

The zero from x to y is $\langle 0 : x \to y,\ 0 : y \to x\rangle$.

It is easy to verify that this construction does indeed produce a bicontext.

Terminology. Any property of orders can be applied to C to mean that every bundle of C has that property. Likewise with a property of dyadic functions to mean that every local composition of C has the property.

3.1.2 DEFINITION. *For $U, V \subseteq C$:*

- $U[x, y] = U \cap C[x, y]$
- $U ; V = \{u ; v \mid u \in U,\ v \in V\}$
- *The* co-Us *are the set* $U^{-} = \{u^{-} \mid u \in U\}$
- *The* symmetric Us, *or* bi-Us, *are the set* $U^{=} = U \cap U^{-}$

We shall write $U ; v$ for $U ; \{v\}$, etc.

3.1.3 PROPOSITION.

- $(U ; V)^{-} = V^{-} ; U^{-}$
- $U^{--} = U$
- $U^{=-} = U^{-=} = U^{=}$
- *If K is a (pre-)sub of C, then so is K^{-}, and $K^{=}$ is a (pre-)bisub.*

3.1.4 PROPOSITION. *If K is a (pre-)sub of C, then so is K^{-}, and $K^{=}$ is a (pre-)bisub.*

3.1.5 PROPOSITION. *Each of the four classes of all small (pre-)(bi)subs is a closure system.*

This means that we can talk about the sub etc. *generated* by a certain subclass of C.

Every object of a context C has a zero loop. The question arises as to whether this can be equal to the identity on the object, and what happens if it is. Obviously, if x has only one loop, this must be both 0 and 1. The next proposition shows that the converse is true: in every other case, 0 and 1 are distinct.

3.1.6 PROPOSITION. *If object x has $x = 0 : x \to x$, then for any y, $C[x, y]$ and $C[y, x]$ are singleton. In particular, $C[x, x] = \{0\}$.*

PROOF: Let $f : x \to y$. Then $f = 1 ; f = 0 ; f = 0$. Similarly for $C[y, x]$. ∎

This motivates:

3.1.7 DEFINITION. *A* zero *(object) is one with* $0 = 1$.

Thus zeroes of a bicontext behave like both initial and terminal objects in an ordinary category and the zeroes of an additive category. Below, we shall see that all zeroes are equivalent in an obvious sense. We shall assume, unless otherwise stated, that a context contains objects other than zeroes. However, there is one particular use for a *zero* context (ie: comprising only zeroes, necessarily symmetric) to represent a class — if X is a class, we shall treat it as the unique zero context with objects X.

3.1.8 DEFINITION. *Let $f : x \to y$ in bicontext C. We define*

- $f^L = f ; f^-$
- $f^R = f^- ; f$

These are the left *and* right ends *of f. They are both loops.*

Immediate properties of the ends are:

- $f^{-L} = f^R,\ f^{-R} = f^L$

- $0^L = 0^R = 0,\ 1^L = 1^R = 1$
- $(\cdot)^L, (\cdot)^R$, qua functions from $C[x,y]$ to $C[x,x]$ and $C[y,y]$ respectively, are continuous.

An important part will be played in the sequel by the next two concepts.

3.1.9 DEFINITION. *A bicontext is* interior *iff every arrow has both ends* ≤ 1.

3.1.10 DEFINITION. $f : x \to y,\ g : y \to z$ fit *iff* f^R *and* g^L *are comparable, or either of* f *or* g *is an object.*

3.1.11 PROPOSITION.

- f, f^- *fit.*
- *If* f, g *fit,* g^-, f^- *do also.*

We now define a series of properties of an arrow based upon the relationship of the ends of the arrow to the identity. In the next definition, θ,... will stand for any of the binary relations $\leq$, $\geq$, $=$ or identically true between loops on an object.

We shall say that arrow f has *signature* $[\theta, \theta']$ iff $f^L\ \theta\ 1$ and $f^R\ \theta'\ 1$. We write $C[\theta, \theta']$ for the set of arrows with this signature (identically true will simply be omitted). These subsets are called *end-properties*.

3.1.12 THEOREM.

(1) $C[\theta, \theta'] < C$. *Furthermore, it is spanning.*

(2) *If* θ *and* θ' *are only either* $\leq$ *or identically true,* $C[\theta, \theta'] \lhd C$.

PROOF:

(1) Any object has all four relations to itself, and therefore possesses every signature. So $C[\theta, \theta']$ is spanning, and it is then trivial that it is closed under taking

source and target. Now

$$
\begin{aligned}
(f\,;g)^L &= f\,;g\,;g^-\,;f^- \\
&= f\,;g^L\,;f^- \\
&\theta\ \ f\,;f^- \\
&= f^L \\
&\theta\ \ 1
\end{aligned}
$$

and each relation is transitive. Similarly for $(f\,;g)^R$ with θ'. Thus $C[\theta, \theta']$ is a sub-category. If $D \nearrow_{C[x,y]} f$ and every member of D has signature $C[\theta, \theta']$, then $f^L = \bigvee_{d \in D} d^L$. For all four relations it is clear that if every d^L has that relation to 1, so has their sup. Similarly for f^R.

(2) Trivial. ∎

The next proposition lists some basic facts about end-properties.

3.1.13 PROPOSITION.

- $C[\theta, \theta']^- = C[\theta', \theta]$
- $C[\theta, \theta]$ *is symmetric.*
- $C[\theta_1, \theta'_1] \cap C[\theta_2, \theta'_2] = C[\theta_1 \cap \theta_2,\ \theta'_1 \cap \theta'_2]$.
- $C[\theta, \theta']^{=} = C[\theta \cap \theta',\ \theta \cap \theta']$

We can now define the end-properties of special interest; for each one we indicate what sort of sub it is.

3.1.14 DEFINITION.

(1) interiors, $\mathrm{Int} = [\leq, \leq]$ $\quad \lhd^{=}$

(2) injections, $\mathrm{Inj} = [=, \leq]$ $\quad <$

(3) projections, $\mathrm{Proj} = [\leq, =] = \mathrm{Inj}^-$ $\quad <$

(4) isos, $\mathrm{Iso} = [=, =]$ $\quad <^{=}$

(5) singulars, $\mathrm{Sing} = [, \leq]$ $\quad \lhd$

(6) totals, $\mathrm{Tot} = [\geq,]$ $<$

(7) expandings, $\mathrm{Exp} = [=,]$ $<$

(8) contractings, $\mathrm{Contr} = [, =] = \mathrm{Exp}^{-}$ $<$

(9) adjunctions, $\mathrm{Adj} = [\geq, \leq]$ $<$

(10) closures, $\mathrm{Clo} = [\geq, \geq]$ $<^{=}$

If we want to refer explicitly to C, we shall write $\mathrm{Int}(C)$ etc. Notice that Int is the only signature that gives another bicontext, which is itself interior. Also, C is interior if and only if $\mathrm{Int}(C) = C$.

When $i : x \to y$ is an iso, we may write $i : x \cong y$. We said above that all zeroes were essentially equivalent: they are in fact all isomorphic, because any arrow between zeroes is itself zero, and therefore has zero ends whence it is an iso. In fact, the unique arrow from a zero to any object is an injection.

We now define a particularly important subclass of loops on an object.

3.1.15 DEFINITION. *For object x, the* parts *of x, $\mathrm{P}x$, is the set $\{f : x \to x \mid f \leq 1\}$. We shall use the term Part to refer variously to the set of all $\mathrm{P}x$ or an arbitrary $\mathrm{P}x$.*

The ends of a part are parts, so $\mathrm{Part} \subseteq \mathrm{Int}$. Indeed, we could characterise interiors as arrows whose ends are parts.

3.1.16 PROPOSITION.

(1) *Part is a bisub.*

(2) *Part(Part)) = Part*

PROOF:

(1) $0 \leq 1$, $x \leq 1$, and if $f, g \leq 1$ so are $f \,;\, g$ and f^{-}. It is obvious that $\mathrm{P}x$ is D-closed, so $\mathrm{P}x \lhd^{=} C$. (2): Immediate. ∎

Terminology. If P is a property of (bi)contexts, we shall say that C is *locally-P* to mean that $\mathrm{Part}(C)$ has P, and that C is *P at x* to mean that $\mathrm{P}x$ has P. Similarly, if $K - \ldots$ is some concept defined relative to a presub $K < C$, when $K = \mathrm{Part}(C)$ we shall talk of a *local-*... or *locally-*.... We may also apply a property of bicontexts to an object x to mean that $\mathrm{P}x$ has the property.

We now give some examples of contexts and bicontexts:

Set.

This is the bicontext comprising all relations between small sets. The ordering on the bundles is set inclusion, composition is relational composition, and reverse is relational inverse. Every bundle is non-empty, with the empty relation as its zero.

Set_0.

This is very similar to Set. It comprises all small *pointed* sets (sets with a distinguished element, 0), and relations $r : x \rightarrow y$ between them for which $0 \notin {!r}$, $0 \notin r!$. We shall see below how to construct Set_0 from Set.

Type.

This is the context comprising all strict maps between small types. The objects are the identity functions, the bundle types are the orders $(x \rightarrow y)$ and composition is ordinary function composition.

Type2.

This is the bicontext Pair(Type). We may often take the liberty of using an arrow of this bicontext to stand for its "forward half" if no confusion will result.

Approximable relations.

Another example would be the class of orders with bottom (0) and approximable relations (see, for example, [1]) $r : x \rightarrow y$ between them.

As with ordinary categories, we can form the product of an I-tuple of (bi)contexts.

- The product of an I-tuple of contexts is the product of the underlying categories with each bundle being the corresponding product order.
- When the component contexts are symmetric, their product is that of the underlying contexts with reverse defined component-wise:

$$f^{-} = \langle f_i^{-} \rangle_{i \in I}$$

We also have a *power* when all the components are equal; in every case we shall employ exactly the same notation as for product categories.

It is easy to check that these products are indeed a context and a bicontext respectively.

Of particular utility is the following special product. Let $\psi : X \to \mathrm{obj}C$ be any function. First construct $C \times X$ (treating X as the zero context) then select the full sub

$$(C \times X)|\{\langle \psi x, x \rangle \mid x \in X\}.$$

We shall call this the *magnification* of C by ψ, and write it as $C \times \psi$. Obviously $C \times \psi$ and X are coincident via $x \longleftrightarrow \langle \psi x, x \rangle$. Moreover, if $K \lhd C$ is spanning, we can also form $K \times \psi$ spanning $C \times \psi$.

3.1.17 PROPOSITION.

(1) $Int(C \times \psi) = \mathrm{Int}(C) \times \psi$

(2) $Part(C \times \psi) = \mathrm{Part}(C) \times \psi$

Now suppose we have classes X, Y and functions $\phi : Y \to X$, $\psi : X \to \mathrm{obj}C$. Define

$$\phi' : Y \to \mathrm{obj}(C \times \psi) : y \mapsto \langle \psi(\phi y), \phi y \rangle.$$

Then clearly the function : $|C \times (\phi \,;\, \psi)| \to |(C \times \psi) \times \phi'|$ that takes

$$\langle \psi(\phi y), y \rangle \mapsto \langle \phi' y, y \rangle$$

and

$$(f : \langle \psi(\phi y), y\rangle \to \langle \psi(\phi y'), y'\rangle) \mapsto (f : \langle \phi' y, y\rangle \to \langle \phi' y', y'\rangle)$$

(well-defined because the f on both sides is in $C[\psi(\phi y), \psi(\phi y')]$) is an isomorphism $: C \times (\phi\,;\psi) \cong (C \times \psi) \times \phi'$.

We shall therefore identify these two contexts and write them commonly as $C \times \psi \times \phi$.

We also have the following relationship between subs and products:

3.1.18 PROPOSITION. *If C is an I-tuple of (bi)contexts, and for each $i \in I$, K_i is some kind of sub of C_i, then $\prod_i(K_i)$ is the same kind of sub of $\prod_i(C_i)$.*

Furthermore, if every $K_i = K$, we shall often refer to K^I simply as K.

Using this machinery, we can define various constructions on contexts. One example is Set_0. Let *drop* be the function that carries every pointed set X to $X \setminus \{0\}$. Then $\mathrm{Set}_0 = \mathrm{Set} \times drop$. We now define some more complicated variations.

It is convenient to introduce the notion of a *square* in a context C. This is a diagram (see Diagram 3.1) with

$$left\,;bottom \leq top\,;right.$$

We might call it "sub-commutative"; if it is properly commutative, ie: equality holds, the square is *exact*. We shall express this situation by saying that $\langle left, right \mid top, bottom\rangle$ is a square. And given square s, $top(s)$ etc. will denote the various edges.

Notice that *left* and *bottom* can be decreased, or *top* and *right* increased, and still retain a square.

Given the corners (x, y, x', y'), the set of squares with those corners can be ordered edgewise. Then the (exact) zero square $\langle 0, 0 \mid 0, 0\rangle$ is obviously bottom in this ordering (if it is non-empty), and if D is a directed set of squares, we have

$$(*) \qquad \begin{aligned} (\bigvee_{d\in D} left(d)) ; (\bigvee_{d\in D} bottom(d)) &= \bigvee_{d\in D} (left(d) ; bottom(d)) \\ &\leq \bigvee_{d\in D} (top(d) ; right(d)) \\ &= (\bigvee_{d\in D} top(d)) ; (\bigvee_{d\in D} right(d)) \end{aligned}$$

so we get a type $\mathrm{Square}(x, y, x', y')$. And if all the squares in D were exact, we would have equality in (*), so the exact squares are a subtype.

Squares can be composed both vertically and horizontally; if two squares have a common edge, the two taken together with that edge removed is also a square, exact if the originals are. For example, Diagram 3.2 shows the horizontal composite of two squares — it is easy to check that

$$\langle left, (bottom ; bottom') \mid (top ; top'), right' \rangle$$

is indeed a square. Moreover, both modes of composition are clearly continuous with respect to the edgewise ordering.

If we are given the *top*, *left* and *bottom* of a square, we could try to complete it with

$$right = top^{-} ; left ; bottom$$

Then

$$\begin{aligned} top ; right &= top ; top^{-} ; left ; bottom \\ &\geq left ; bottom \end{aligned}$$

provided *top* is total.

Similarly, if *left* is missing, setting $left = top ; right ; bottom^{-}$ yields a square if *bottom* is singular. Thus:

3.1.19 PROPOSITION.

(1) *right can be filled in if top is total, exactly if it is expanding.*

(2) *left can be filled in if bottom is singular, exactly if it is contracting.*

Corresponding assertions can be made about *top* and *bottom*, but they are less important for our purposes.

Now let

$$end : |C| \to \mathrm{obj}C^2 : (f : x \to y) \mapsto \langle x, y \rangle.$$

If we magnify C^2 with *end*, we will get a context whose objects are arrows of C, and whose arrows are pairs of C-arrows between the sources and the targets. We therefore define the *arrow context* of C, $\mathrm{Arr}(C)$, to be $C^2 \times end$.

Now a square can be viewed as a member of $\mathrm{Arr}(C)$ in two ways: either from left to right, or from top to bottom. Each of these views, together with the appropriate kind of composition, effectively makes the squares a sub of $\mathrm{Arr}(C)$. We shall call them $\mathrm{Hor}(C)$ or $\mathrm{Vert}(C)$. The exact arrows comprise further subs $\mathrm{Hor}_{=} \lhd \mathrm{Hor}$ and $\mathrm{Vert}_{=} \lhd \mathrm{Vert}$.

If C is symmetric, what is the appropriate notion of reversal of a square? We could reverse all the edges (and rotate through π), but this would yield a square iff the original were exact. It seems preferable to reverse it either horizontally or vertically, ie: with respect to either Hor or Vert. In the first case we reverse only *top* and *bottom*, in the second only *left* and *right*. We shall say that a square is *horizontally* or *vertically symmetric* when its corresponding reversal is also a square. It follows that the horizontally symmetric squares are a bisub of Hor, the vertically symmetric ones a bisub of Vert.

3.1.20 PROPOSITION. *A square is*

(1) *vertically symmetric if its top is singular and its bottom total.*

(2) *horizontally symmetric if its left is co-total and its right is co-singular.*

PROOF:

(1) We have

$$left(*)\ bottom \leq top\,;right \Rightarrow bottom^{-}\,;left^{-} \leq right^{-}\,;top^{-}$$

$$\Rightarrow bottom\,;bottom^{-}\,;left^{-}\,;top \leq bottom\,;right^{-}\,;top^{-}\,;top$$

$$\Rightarrow left^{-}\,;top \leq bottom\,;right^{-}$$

(2) From (*) continue with ...

$$\Rightarrow right\,;bottom^{-}\,;left^{-}\,;left \leq right\,;right^{-}\,;top^{-}\,;left$$

$$\Rightarrow right\,;bottom^{-} \leq top^{-}\,;left$$

∎

3.1.21 COROLLARY. *A square is vertically symmetric if its top and bottom are adjunctions, and horizontally symmetric if its left and right are co-adjunctions.*

3.2 STRONG AND REGULAR INTERIORS

In the bicontext Set, the interiors are (1,1) relations. These have some nice properties not possessed by interiors in general. This is particularly evident in the case of parts. In this section we look at particular kinds of interiors, with properties that bring them closer to interiors in Set.

Let us fix some bicontext C for this section.

An interior has both ends ≤ 1, so, for $f \in \mathrm{Int}$, $f\,;f^{-}\,;f \leq f$. We define strong interiors as those for which equality holds.

3.2.1 DEFINITION. *Interior f is* strong *iff $f;f^{-};f = f$ (equivalently $f;f^{-};f \geq f$).*

3.2.2 PROPOSITION.

(1) *Interior f is strong iff f^{-} is.*

(2) *The strong interiors are a subtype of any given bundle.*

(3) *Every injection (and therefore projection) is strong.*

PROOF:

(1) f strong implies $f ; f^- ; f = f$ implies $f^- ; f ; f^- = f^-$ implies f^- strong.

(2) Let D be a directed set of strong interiors in $C[x, y]$ with sup f. Then

$$f ; f^- ; f = \bigvee_{d \in D} (d ; d^- ; d) = \bigvee_{d \in D} d = f$$

(3) When $f \in \mathrm{Inj}$, $f ; f^- ; f = f^L ; f = f$, so f is strong. Likewise for Proj. ∎

Unfortunately, the strong interiors do not necessarily form a sub; we shall see a counter-example for this in Chapter 6. However, the strong parts are important; we shall refer to them again later, but for the moment we can characterise them by:

3.2.3 PROPOSITION. *$p \in \mathrm{P}x$ is strong iff $p = p^- = p ; p$*

PROOF:

$$\begin{aligned} p \text{ strong } &\Rightarrow p = p ; p^- ; p \leq \{p^-, \ p ; p\} \\ &\Rightarrow p = p^- = p ; p \quad (\text{since } p ; p \leq p) \\ &\Rightarrow p ; p^- ; p = p ; p ; p = p \\ &\Rightarrow p \text{ strong.} \end{aligned}$$

∎

Also, if $p ; q$ is strong ($p, q \in \mathrm{P}x$), it is the inf of $\{p, q\}$ amongst the strong parts of x. For clearly $p ; q \leq p, q$, and r strong $\leq p, q$ implies $r = r ; r \leq p ; q$.

A forgetful functor $F : C \to C'$ will reflect strong parts, ie: Fp strong implies p strong. In particular, if C' has all parts strong (eg: Set), then so does C.

Although the strong interiors do not form a sub, we can, if we work in a pre-compact bicontext, define a different kind of interior which constitutes a bisub.

So for the remainder of this section, C will be pre-compact.

3.2.4 DEFINITION. *Let $k \in C$. Then k is:*

(1) L-regular *when*, $\forall f, g \in C,\ f \leq k\,;g \Rightarrow f \leq k^L\,;f$

(2) R-regular *when* k^- *is L-regular*

(3) regular *when it is both L- and R-regular*

3.2.5 PROPOSITION. *If k is L-regular, then $k \leq k^L\,;k$*

It is now immediate that a regular interior is strong. The next theorem shows that the regular interiors do indeed form a bisub. We prove it by way of a Lemma. We shall write RegInt for the set of regular interiors.

3.2.6 LEMMA. *For any objects x, y, $\{k : x \to y \mid k \text{ is regular}\}$ is a subtype of $C[x, y]$.*

PROOF:

(a) We have

$$\begin{aligned} f \leq 0\,;g &\Rightarrow f \leq 0 \\ &\Rightarrow f = 0 \\ &\Rightarrow 0^L\,;f \geq f \end{aligned}$$

so 0 is L-regular and therefore regular by symmetry.

(b) Let $D \nearrow_{C[x,y]} d_0$ with every $d \in D$ regular. Let $f \leq d_0\,;g$. Clearly the set $D\,;g = \{d\,;g \mid d \in D\} \nearrow d_0\,;g$, so by pre-compactness of C, there is $D_f \nearrow f$ dominated by $D\,;g$. That is to say, for $f' \in D_f$, $f' \leq d\,;g\ \exists d \in D$. It follows that $d^L\,;f' \geq f'$, so $d_0^L\,;f' \geq f'$, whence $d_0^L\,;f \geq f$. Thus d_0 is L-regular.

R-regularity is similar. ∎

3.2.7 THEOREM. RegInt *is a spanning bisub of C.*

PROOF: RegInt is obviously closed under $(\cdot)^-$. By 3.2.6, it contains every 0. Also, every object is an interior, obviously regular, and so is in RegInt. Thus RegInt is spanning. Using 3.2.6 again, it remains only to show that RegInt is closed under composition. So let $k_1, k_2 \in \text{RegInt}$ and let $f \leq k_1 ; k_2 ; g$. Then $k_1^L ; f = f$ and $k_1^- ; f \leq k_1^R ; k_2 ; g \leq k_2 ; g$ implies $k_2^L ; k_1^- ; f = k_1^- ; f$, which implies $(k_1 ; k_2)^L ; f = k_1 ; k_2^L ; k_1^- ; f = k_1^L ; f = f$. So $k_1 ; k_2$ is L-regular. Likewise $k_2^- ; k_1^-$ is L-regular, so $k_1 ; k_2$ is R-regular. We already know that $k_1 ; k_2$ is an interior. ∎

Thus we have:

$$\text{RegInt} \subseteq \text{strong Ints} \subseteq \text{Int}$$

$$\text{RegInt} \trianglelefteq^{=} \text{Int}$$

3.3 Functors over Bicontexts

3.3.1 Definition. *Let C, C' be contexts.*

(1) *A* pre-functor from C to C' *is a function $F : C \to C'$ which is a pre-functor of the underlying category.*

(2) *A* functor *is a pre-functor F for which every $F[x, y]$ is continuous.*

When C, C' are symmetric, $F : C \to C'$ is a functor (of bicontexts) when it is also symmetric. We shall write $|F|$ for the underlying function of $F : |C| \to |C'|$ and obj$F$ *for the restriction of F to :* obj$C \to$ objC'. *Also, if $U \subseteq C$ and R is a binary relation on C, FU will mean $\{Ff \mid f \in U\}$ and FR will mean $\{\langle Ff, Fg\rangle \mid f\ R\ g\}$.*

Additionally, we have: A pre-functor is *spanning* when every object in C' is the image of one in C. If F is polyadic with C an I-tuple of contexts, it *takes* $\langle K_i \rangle_i$ (each $K_i < C_i$) *to* $K' < C'$ when $F(\prod_i K_i) \subseteq K'$. If each $K_i \trianglelefteq^{=} C_i$, we may say that F *cuts down to* $\langle K_i \rangle_i \to K'$. If K is common to all the C_i and C', and F takes $\langle K \rangle_i$ to K, F is *within* K. F is K*-full* when every F-bundle is onto the corresponding K-bundle (omit $K = C$).

Note that F is *not* in general a functor of the underlying category; this is true only in the case of an exact functor (see below). All we know is that $F(f ; g)$ and $Ff ; Fg$ are parallel.

Terminology. Any property of maps can be applied to a pre-functor to mean that every bundle has that property.

Given a polyadic (pre-)functor $F : \langle C_i \rangle_{i \in I} \to C'$, $J \subseteq I$ and *idempotent loops* u_i for $i \in I$ (ie: each $u_i \,;\, u_i = u_i$), we can *specialise* F to a (pre-)functor

$$F_{u,J} : \langle C_j \rangle_{j \in J} \to C' : f \mapsto F(u[J/f]).$$

And if every $C_i = C$, we can *diagonalise* F to the (pre-)functor

$$F_\Delta : C \to C' : f \mapsto F(\langle f \rangle_i).$$

We can mimic ordinary category theory and introduce a notion of natural transformation between pre-functors. However, there does not appear to be much use for a direct transfer of the concept; rather, we shall define a weakened form which is "sub-commutative" in the same way that squares are.

Let C be a context and I an ordinary (small) category. Let $\mathrm{Pref}(I, C)$ be the class of pre-functors from I to the underlying category of C, and define the function

$$ext : \mathrm{Pref} \to \mathrm{obj}C^{\mathrm{obj}I} : F \mapsto \langle F_i \rangle_i.$$

The the context C^I is defined as $C^{\mathrm{obj}I} \times ext$. We shall refer to the arrows as *transformations*.

3.3.2 DEFINITION. *$t : F \to G$ in C^I is a* natural transformation *(NT) when, for any $f : i \to j \in I$, $\langle t_i, Gf \mid Ff, t_j \rangle$ is a square (see Diagram 3.3)*

These squares are the components *of the NT. t is* exact *when all its components are (write $t : F \xrightarrow{=} G$ in this case).*

Using the properties of squares, it is straightforward to verify that the NTs comprise a sub $\mathrm{NT}(I, C)$, as do the exact ones ($\mathrm{NT}^{=}(I, C)$). $\mathrm{Diag}(I, C)$, the context of *diagrams* over C, will be NT$|\{$functors $: I \to C\}$. We shall usually denote a diagram by some version of $[\![\cdot]\!]$.

If we identify an object $x \in C$ with the constant functor $\lambda f.x : I \to C$, then a NT from x to y (necessarily exact) is simply an arrow $t : x \to y$ in C, and vice versa.

If C is symmetric, so is C^I, and to say that a NT is *symmetric* means relative to C^I.

There is a another view of C^I. Each $t : F \to G \in C^I$ determines, and is determined by, the pre-functor

$$(f : i \to j) \mapsto (\langle Ff, Gf\rangle : t_i \to t_j) : I \to \mathrm{Arr}(C).$$

If t is natural, this pre-functor is into $\mathrm{Hor}(C)$, and if F and G are diagrams, it is also one. That is, a NT can be treated as a diagram in $\mathrm{Arr}(C)$.

3.3.3 Definition. *Let $F : C \to C'$ be a pre-functor, and let θ stand for any one of $\leq, \geq, =$.*

(1) *For composable arrows $f, g \in C$, F is θ* on *$\langle f, g\rangle$ when $F(f\,;g)\ \theta\ Ff\,;Fg$.*

(2) *Let R be any binary relation on C. Then F is R-θ iff it is θ on every $\langle f, g\rangle \in R$.*

(3) *Given an I-tuple C of bicontexts and an I-tuple R of binary relations with each R_i on C_i, we define $S = \prod R$ on $\prod C$ by $\langle f, g\rangle \in S$ iff each $\langle f_i, g_i\rangle \in R_i$.*

Terminology. The cases $\leq, \geq, =$ of θ are referred to as *lower, upper, exact* respectively.

The following special relations are particularly important:

(1) For $K < C$, *R-θ on K* will mean $(R \cap K^2)$-θ.

(2) For $K, J < C$, *K, J-θ* will mean $(K \times J \cup J \times K)$-$\theta$. Omit $J = C$ or $K = C$.

(3) If $X \subseteq \mathrm{obj}C$, *pivot-$\theta$ on X* will mean R-θ, where

$$R = \{\langle f : x \to y,\ g : y \to z\rangle \mid y \in X\}$$

When C is symmetric we have also:

(4) K-*end* is the relation $\{\langle f, f^-\rangle, \langle f^-, f\rangle \mid f \in K\}$

(5) K-*fit* is the relation $\{\langle f, g\rangle \mid f, g \in K \text{ fit}\}$

In (4) and (5) omit $K = C$.

It is clear that, if $K <^= C$, a functor which is θ on K is end-θ on K.

Henceforth in this section, all contexts will be symmetric unless otherwise specified.

Notation. When F is end-exact, $F(f^L) = (Ff)^L$, so we can unambiguously write Ff^L; likewise Ff^R.

3.3.4 PROPOSITION.

(1) *If $R \subseteq S$ are both binary relations on C, and pre-functor F is S-θ, it is also R-θ.*

(2) *If F is K, J-θ, it is also K^-, J^--θ.*

(3) *Let $F : C \to C'$, $G : C' \to C''$ be respectively R-θ, S-θ such that $f\ R\ g \Rightarrow Ff\ S\ Fg$. Then their composite is an R-θ (pre-)functor.*

PROOF:

(1) Obvious

(2) $F(j^- ; k^-) = F(k ; j)^-\ \theta\ (Fk ; Fj)^- = Fj^- ; Fk^-$

(3) The composite is a pre-functor of the underlying category of C, and is obviously a functor if F, G are. Then, for $f\ R\ g$, we have

$$GF(f ; g)\ \theta\ G(Ff ; Fg)\ \theta\ GFf ; GFg$$

— the result follows by transitivity of θ. ∎

In ordinary category theory, the image of a functor is a sub-category. For contexts this only happens for exact functors.

3.3.5 PROPOSITION. *Let $F : C \to C'$ be exact on $K < C$. Then $FK < C'$. Also, K symmetric implies FK is, and if K is a bisub FK is* provided F is strict.

The applications of this proposition to the special relations described above are particularly important.

Examples.

(1) The simplest example of a functor is an inclusion functor $\subseteq : K \lessdot^{=} C$. It is both strict and exact, and full precisely when K is full.

(2) Another naturally occurring example is the collection of polyadic projection functors $(\cdot)_i$ from a sequence of bicontexts to each component, and their corresponding monadic projection functors over the product. They are all exact, linear and full.

(3) An isomorphism between bicontexts is an exact bijective bimonotonic functor

.

(4) If C, D are I-tuples of bicontexts and F is an I-tuple of (pre-)functors with each $F_i : C_i \to D_i$ $(i \in I)$, the product $\prod F$ is also a (pre-)functor.

Moreover, if R is an I-tuple of binary relations with each R_i on C_i and each F_i is R_i-θ, then $\prod F$ is $\prod R$-θ.

(5) Any function between classes determines a unique (and exact) functor between the corresponding zero-contexts.

(6) Any pre-functor from C to C is O-exact, where

$$O = \{\langle x, f : x \dashrightarrow y\rangle,\ \langle f : x \dashrightarrow y, y\rangle\}$$

We now introduce the important concept of a K-functor.

3.3.6 DEFINITION. *If C is an I-tuple of bicontexts, with $K < C_i$ for every $i \in I$, and $F : C \to C'$, then F is a K-functor when it is:*

- *within K*

- *upper*
- *K-exact (equivalently K-lower)*

The specialisation of a K-functor to loops *in* K is also one, as is its diagonalisation (if appropriate). Moreover, a K-functor is also a K^--functor.

We now define some special types of functor. Thread functors will appear later, and forgetful functors, as in ordinary category theory, arise commonly in passing from a bicontext to some underlying one with less structure that the former is based upon.

3.3.7 DEFINITION. *A functor between bicontexts is:*

(1) *A* thread functor *when it is lower and fit-exact.*

(2) forgetful *when it is exact and every bundle is linear bimonotonic.*

(3) *An* embedding *when it is full forgetful and (1,1) on objects.*

Obviously a thread functor is end-exact. Proposition 3.3.4 tells us that the properties of being a K-functor and a thread functor are preserved by functional composition.

When there is a forgetful functor from C to C', we say that C is *C'-based*, and if the functor happens to be full, that C is *C'-like*.

Forgetful $F : C \to C'$ may be *abstract*, which means that, if $i' : Fx \cong y' \in C'$, then there is an $i \in C$ with $Fi = i'$. Furthermore, we have

3.3.8 PROPOSITION. *If F is forgetful and $i : x \to y$ with $Fi : Fx \cong Fy$, then i is an iso.*

PROOF: $F(i\,;i^-) = Fx,\ F(i^-\,;i) = Fy$

Examples.

(1) The prime example of a forgetful functor is an inclusion functor $\subseteq : K \lhd^{=} C$. This is in fact an embedding; conversely, the image of an embedding is a full bisub. In case the inclusion is also abstract, we say that K is abstract.

(2) An isomorphism is an onto embedding.

(3) The functor

$$(f : x \to y) \mapsto (f : drop(x) \to drop(y)) : \mathrm{Set}_0 \to \mathrm{Set}$$

is forgetful, as indeed is any first-component projection $: C \times \psi \to C$. These are full, so every $C \times \psi$ is C-like. By abuse of notation, we shall call the functor ψ as well. Thus we have $drop : \mathrm{Set}_0 \to \mathrm{Set}$.

Example (3) has a converse, yielding a representation for C-like bicontexts:

3.3.9 THEOREM. *Bicontext C is C'-like iff it is a magnification of C'.*

PROOF: It remains to prove the "only if". Let $F : C \to C'$ be full forgetful. Then

$$(f : x \to y) \mapsto (Ff : \langle Fx, x\rangle \to \langle Fy, y\rangle) : C \to C' \times \mathrm{obj}\, F$$

is clearly an isomorphism. ∎

Moreover, it is easy to see that $\psi : C' \times \psi \to C'$ is an embedding precisely when ψ is $(1,1)$.

Obviously any functor preserves part-hood, so any functor from C to C' takes $\mathrm{Part}(C)$ to $\mathrm{Part}(C')$. It is clear that such a restriction of a forgetful functor is still forgetful.

Once again, if P is some property of functors, we shall say that F is *locally-P* to mean that F restricted to Part has P.

Later, we shall use the notion of a disjunctive functor, which we now define.

3.3.10 DEFINITION. *Pre-functor $F : C \to C'$ is* disjunctive *iff given $x \in \mathrm{obj}C$, every $q \in \mathrm{P}Fx$ has*

$$q = \bigvee\{Fp \mid p \in \mathrm{P}x \ \& \ Fp \leq q\}.$$

The action of functors on squares.

In (3.1) we introduced the idea of a square. Now we look at how functors affect squares, in particular NTs.

Let R be a binary relation on bicontext C. Let I be a category. We make the following definitions:

- Pre-functor $P : I \to C$ *satisfies* R (is an R-pre-functor) when, given $f : i \to j,\ g : j \to k\ \in I,\ Pf\ R\ Pg$.
- A square in C *satisfies* R (is an R-square) when both *left* R *bottom* and *top* R *right*.
- A NT from I to C *satisfies* R (is an R-NT) when each of its component squares does.

We can extend R to $C^2 \times |C|$ and $C^{\mathrm{obj}I} \times \mathrm{Pref}(I, C)$ as $R^2 \times |C|^2$ and $R^{\mathrm{obj}I} \times \mathrm{Pref}(I, C)^2$ respectively. These relations then restricted to $\mathrm{Hor}(C)$, $\mathrm{Vert}(C)$ and $\mathrm{NT}(I, C)$ yield respectively R_{Hor}, R_{Vert} and R_{NT}.

Now let $F : C \to D$ be a (pre-)functor to bicontext D. We can extend F to $C^2 \times |C| \to D^2 \times |D|$ and $C^{\mathrm{obj}I} \times \mathrm{Pref}(I, C) \to D^{\mathrm{obj}I} \times \mathrm{Pref}(I, D)$ as $F^2 \times |F|$ and $F^{\mathrm{obj}I} \times$ (post composition with F).

Clearly the former cuts down to $\mathrm{Arr}(F) : \mathrm{Arr}(C) \to \mathrm{Arr}(D)$ and the latter to $F^I : C^I \to D^I$. The question then arises as to whether they also cut down to $\mathrm{Hor}(C) \to \mathrm{Hor}(D)$, $\mathrm{Vert}(C) \to \mathrm{Vert}(D)$, $\mathrm{NT}(I, C) \to \mathrm{NT}(I, D)$ etc. In general the answer is no; the best we can do is:

3.3.11 PROPOSITION. *Let F be R-exact. Then*

(1) *If diagram $[\![\cdot]\!] : I \to C$ satisfies R , then $F[\![\cdot]\!]$ is a diagram in D.*

(2) *F applied (edgewise) to an R-square in C is a square in D, exact if the original is.*

(3) *If $t : P \to P'$ is an R-NT in C^I, then $Ft : FP \to FP'$ is a NT in D^I, exact if t is.*

Lifting.

If $V : C \to C'$ is a functor, it may serve to relate functors over C to functors over C', and this relationship is of special interest when V is forgetful. We shall first define the relationship, then catalogue some of the salient functor properties which are preserved by lifting via a forgetful functor.

3.3.12 DEFINITION. *Given functors $V : C \to C'$, $F : C^I \to C$, $F' : C'^I \to C'$ (I is an index set), we say that F* lifts *F'* (via *V) when, for any $f \in C^I$,*

$$VFf = F'Vf$$

where Vf is short for $\langle Vf_i \rangle_{i \in I}$

3.3.13 THEOREM. *With the same notation, let V be forgetful and let R be a binary relation on C. Then*

(1) *If F' is any of bimonotonic, definite, monolinear, polylinear, monostrict or polystrict, so is F.*

(2) *If F' is VR-lower, upper, exact, F is likewise relative to R*

(3) *If $F'V$ is disjunctive, so is F*

(4) *If F', V are both projective, so is F.*

PROOF:

(1) Bimonotonic:

$$\begin{aligned} Ff \leq Fg &\Rightarrow VFf \leq VFg \\ &\Rightarrow F'Vf \leq F'Vg \\ &\Rightarrow Vf \leq Vg \\ &\Rightarrow f \leq g \end{aligned}$$

Definite: $0 < Ff$ implies $0 = V0 < VFf$, so the "bimonotonic" part of definiteness goes through as above. And

$$\begin{aligned} Ff = 0 &\Rightarrow F'Vf = VFf = 0 \\ &\Rightarrow Vf_i = 0 \quad \exists i \in I \\ &\Rightarrow f_i = 0. \end{aligned}$$

Monolinear: Let $A \subseteq C^I$. Then

$$\begin{aligned} VF\bigvee A &= F'V\bigvee A \\ &= F'\bigvee VA \\ &= \bigvee F'VA \\ &= \bigvee VFA \\ &= V\bigvee FA \end{aligned}$$

Polylinear: If the members of A above agree on all but one component, so do the members of VA, so the argument goes through again. The case where the component in focus has $0 = \bigvee \emptyset$ is covered by the argument for polystrictness below.

Monostrict: $VF0 = F'V0 = F'V0 = F'0 = 0$

Polystrict: $VF(..., 0, ...) = F'(V..., 0, V...) = 0$

(2) Let $f\ R\ g$ be composable. Then

$$
\begin{aligned}
VF(f\,;g) &= F'V(f\,;g) \\
&= F'(Vf\,;Vg) \\
&\leq, \geq, = F'Vf\,;F'Vg \quad \text{since } Vf\ VR\ Vg \\
&= VFf\,;VFg \\
&= V(Ff\,;Fg)
\end{aligned}
$$

(3) Let $x \in \mathrm{obj}C$, $q \in \mathrm{P}Fx$. Let $Y = \{Fp \mid p \in \mathrm{P}x \ \&\ Fp \leq q\}$. Clearly $Y \leq q$. Then

$$
\begin{aligned}
VY &= \{VFp \mid p \in \mathrm{P}x \ \&\ Fp \leq q\} \\
&= \{F'Vp \mid p \in \mathrm{P}x \ \&\ F'Vp \leq Vq\}
\end{aligned}
$$

So

$$
\begin{aligned}
Y \leq y &\Rightarrow VY \leq Vy \\
&\Rightarrow \bigvee VY = Vq \leq Vy \\
&\Rightarrow q \leq y
\end{aligned}
$$

Thus $q = \bigvee Y$

(4) Let $X = \{f : x \to y \mid Ff \leq g\} \neq \emptyset$. Let f_0 be top in $\{f \mid Vf \leq f_0'\}$, where f_0' is top in $\{f' \mid F'f' \leq Vg\}$ ($Ff \leq g$ implies $VFf = F'Vf \leq Vg$, so the last set is non-empty, whence f_0' exists, so that $Vf \leq f_0'$ making the second set non-empty, whence f_0 exists). Then

$$
\begin{aligned}
Vf_0 \leq f_0' &\Rightarrow F'Vf_0 \leq Vg, \\
&\Rightarrow VFf_0 \leq Vg \\
&\Rightarrow Ff_0 \leq g.
\end{aligned}
$$

So $f_0 \in X$, whence

$$
f \in X \Rightarrow VFf \leq Vg
$$

$$\Rightarrow F'Vf \leq Vg$$

$$\Rightarrow Vf \leq f_0'$$

$$\Rightarrow f \leq f_0$$

∎

Furthermore, suppose $\psi : X \to \mathrm{obj}C$ and that $F : C^I \to C$ is a (pre-)functor. If $G : X^I \to X$ is such that $\psi(Gx) = F(\psi^I x)$ for every $x \in X^I$, then G extends canonically to an I-ary (pre-)functor on $C \times \psi$ that lifts F, viz:

$$G(f : \langle x, \psi^I x\rangle \to \langle y, \psi^I y\rangle) = Ff : \langle Gx, \psi Gx\rangle \to \langle Gy, \psi Gy\rangle$$

Translators.

Sometimes we encounter functions defined only on a single bundle of a bicontext or which, for other reasons, are not quite functors. The following concepts may then be convenient:

3.3.14 DEFINITION. *Let C, C' be bicontexts with $x, y \in C$, $x', y' \in C'$*

(1) *A continuous function $T : C[x, y] \to C'[x', y']$ is a* translator *(*from x, y to *x', y'). In case $x = y$, $x' = y'$, it is* unitary *iff $Tx = x'$.*

(2) *A translator from x, y to x, y is a (x, y)-*recursor*. When $x = y$, it is a recursor* on x.

Recursors are important because of their least fixed points:

3.3.15 PROPOSITION. *Let R be an (x, y)-recursor. Then*

(1) $\mathrm{fix}\, R : x \to y$.

(2) *If $x = y$, $\mathrm{fix}\, R \in \mathrm{P}x$, and if R is symmetric, so is $\mathrm{fix}\, R$.*

3.4 K-ALIGNMENT OF A FUNCTOR

In this section, let C, C' be bicontexts with $K <^{=} C$ pre-compact and spanning, and let $F : C \to C'$ be a functor. If either (a) C' is linear almost-complete, or (b) K is projective, we can define a new functor from F which is essentially the same as F but better behaved with respect to K.

3.4.1 DEFINITION. *Define* $F_K : C \to C'$ *by* $F_K f = \bigvee F f_K$, *where* $f_K = (f] \cap K$.

(This is a good definition because $F f_K$ is either (a) $\leq Ff$ or (b) directed or empty.)

3.4.2 THEOREM. F_K *is a functor from* C *to* C' *which agrees with* F *on* K.

PROOF: Since K is spanning, every $1 \in f_K$, so $F_K 1 = F1 = 1$. Obviously $F_K f$ is parallel to Ff so F_K is a pre-functor. Next, because K is symmetric, $f^-{}_K = f_K{}^-$ (write f_K^-), so $F f_K^- = (F f_K)^-$, whence $F_K f^- = (F_K f)^-$. And for $k \in K$, k is top in k_K, so $F_K k = Fk$. It remains to show every $F_K[x,y]$ continuous. First,

$$f \leq g \Rightarrow f_K \subseteq g_K \Rightarrow F_K f \leq F_K g,$$

so F_K is monotonic. Now let $D \nearrow_{C[x,y]} d$. For $k \in d_K$, let, by pre-compactness, $K_k \nearrow_K k$, with K_k dominated by D. Then

$$d_K = \{\bigvee K_k \mid k \in d_K\}$$

implies

$$F d_K = \{\bigvee F K_k \mid k \in d_K\}$$

implies

$$\begin{aligned} F_K d &= \bigvee \bigcup \{F K_k \mid k \in d_K\} \\ &= \bigvee F(\bigcup \{K_k \mid k \in d_K\}) \end{aligned}$$

But $\bigcup\{K_k \mid k \in d_K\} \subseteq \bigcup\{d_K \mid d \in D\}$. So

$$F_K d \leq \bigvee \bigcup \{F d'_K \mid d' \in D\}$$

$$= \bigvee\{\bigvee Fd'_K \mid d' \in D\}$$

$$= \bigvee F_K D$$

∎

3.4.3 THEOREM.

(1) *If F is upper on K, F_K is upper.*

(2) *If $K \subseteq RegInt$ and F is lower on K, then F_K is K-lower.*

PROOF:

(1)

$$f_K \,;\, g_K \subseteq (f \,;\, g)_K$$

so

$$\begin{aligned} F_K(f \,;\, g) &= \bigvee F(f \,;\, g)_K \\ &\geq \bigvee F(f_K \,;\, g_K) \\ &\geq \bigvee Ff_K \,;\, Fg_K \\ &= \bigvee Ff_K \,;\, \bigvee Fg_K \end{aligned}$$

by either (a) linearity or (b) strictness and continuity of ;

$$= F_K f \,;\, F_K g$$

(2) $k' \leq k \,;\, f \;\Rightarrow\; k^- \,;\, k' \leq k^R \,;\, f \leq f$, and by regularity that $k^L \,;\, k' = k'$, ie: that $k' = k \,;\, (k^- \,;\, k')$

Thus $(k \,;\, f)_K = k \,;\, (f_K)$ and we have

$$\begin{aligned} F_K(k \,;\, f) &= \bigvee F(k \,;\, f_K) \\ &\leq \bigvee Fk \,;\, Ff_K \\ &= Fk \,;\, \bigvee Ff_K \end{aligned}$$

by either ... as above

$$= F_K k \,;\, F_K f$$

∎

Moreover, if F is polyadic (of arity I), $K = K_0^I$, and F is within K_0, then every k_K has k as top, so $F_K k = Fk \in K_0$ and F_K is within K_0. In this case we shall write F_{K_0} for F_K.

3.4.4 COROLLARY. *If $K \subseteq RegInt$ and F is exact on K, F_K is upper and K-exact. If additionally F is within K, F_K is a K-functor.*

3.5 JOINS AND SUMS

In this section we shall look at some constructions in bicontexts that are akin to biproducts in Ab-categories. Indeed, the definition of sum is *formally* the same as that of biproduct in [16], if the group operation be replaced by $\vee$.

First, fix some bicontext C and an index-set I.

3.5.1 DEFINITION. *Functor $F : C^I \to C$ is a* prejoin *on C when, for any object $x \in C^I$ and any $i \in I$, there are* injections

$$\eta_{x,i} : x_i \to Fx.$$

Notation.

In the situation of 3.5.1, if $f : x \to y \in C^I$ and $g : Fx \to Fy$ (see Diagram 3.4) we define

$$f\_i = \eta_{x,i}^- \,;\, f_i \,;\, \eta_{y,i},$$

$$g|i = \eta_{x,i} \,;\, g \,;\, \eta_{y,i}^-$$

$$g| = \langle g|i \rangle_i \in C^I.$$

When there is no risk of confusion, we may abbreviate $\eta_{x,i}$ variously to η_x, η_i or just η.

The following facts are immediate.

(1) $(\cdot)|$ is a unitary translator from Fx, Fy to x, y with

$$(g \,;\, g')| \geq g| \,;\, g'|$$

Indeed, $(\cdot)|i$ is an upper functor : $C \times \mathrm{obj}F \to C^I$.

(2) $(\cdot)\_i$ is a (non-unitary) translator from x_i, y_i to Fx, Fy, with

$$(f\,;\,f')\_i = f\_i\,;\,f'\_i$$

(3) $g|\_i \leq g$

(4) $f\_i|i = f_i$

(5) $x\_i = \eta^R_{x,i} \leq Fx$

It is not possible similarly to convert $(\cdot)\_i$ into a functor, because it does not even preserve objects, although it is "exact".

3.5.2 DEFINITION. *A* join *on C is a prejoin, $+$, which satisfies, for $f : x \to y \in C^I$, $+f \geq f\_i$, $\forall i \in I$. It is a* sum *when it further satisfies $+f = \bigvee_{i \in I} f\_i$.*

Write $f_1 + \cdots + f_n$ for $+(f_1, \ldots, f_n, 0, \ldots)$.

Actually, the join condition forces the η to be injections.

3.5.3 PROPOSITION. *Let $+$ be a join. Then*

(1) $(+f)| \geq f$

(2) *$+$ is a sum iff, for any $g : +x \to +y$, $+(g|) \leq g$*

(3) *If $+$ is a sum, $+f = \bigvee\{f\_i \mid f_i \neq 0\}$ (whence $+0 = 0$)*

(4) $f\_i = +\,0[i/f_i]$

PROOF:

(1) $+f\ |i \geq f\_i|i = f_i$

(2) If $+$ is a sum, then $+\,g| = \bigvee_i g|\_i \leq g$. Conversely, let g dominate every f_i. Then $g| \geq f\_i| = f$, whence

$$g \geq +\,g| \geq +f$$

(3) Every $0\_i = 0$

(4) Immediate from (3). ∎

It follows that, when $+$ is a sum, $+(+f\,|) = +f$, $(+\,g|)| = g|$ and $g|$ is the largest $f : x \to y$ with $+f \leq g$. Thus $+$ is projective, hence monolinear. Also, equality in (1) is equivalent to $+$ being bimonotonic.

Of particular interest are sums for which equality holds in 3.5.3(2) above for all *parts* g. The next theorem tells the full story, but first we prove:

3.5.4 LEMMA. *If join $+$ is disjunctive, $+\,q| \geq q$, $\forall q \in \mathrm{P}{+}x$.*

PROOF: Let $q \in \mathrm{P}{+}x$. Then

$$\begin{aligned} +\,q| &\geq \bigvee\{+(+p\,|) \mid p \in \mathrm{P}x \;\&\; +p \leq q\} \\ &= \bigvee\{+p \mid p \in \mathrm{P}x \;\&\; +p \leq q\} \\ &= q \end{aligned}$$

∎

3.5.5 THEOREM. *If $+$ is a sum, the following are equivalent:*

(1) *$+$ is locally-full*

(2) *Every $q \in \mathrm{P}{+}x$ is $+p$ for some $p \in \mathrm{P}x$*

(3) *$+$ is disjunctive.*

(4) *Equality holds in 3.5.3(2) for all parts q.*

PROOF:

(1) implies (2): Let $q \in \mathrm{P}{+}x$, with $q = +p$. Then by 3.5.3(3), $q = +(q|)$. Since $(\cdot)|$ preserves part-hood, $q| \in \mathrm{P}x$.

(2) implies (3): $q \in \{+p \mid p \in \mathrm{P}x \;\&\; +p \leq q\}$

(3) implies (4): Immediate from the previous Lemma.

(4) implies (1): A fortiori

In this situation, we shall normally use the term "disjunctive" to describe the sum.

We now show that a sum must be a lower functor.

3.5.6 THEOREM. *A sum is a lower functor.*

PROOF: Let $+$ be a sum on C. Then, if $f, g \in C^I$ with $f : x \to y$ and $f' : y \to z$, we have

$$\begin{aligned} +f\,;f' &= \bigvee_i (f\,;f')\_i \\ &= \bigvee_i (f\_i\,;f'\_i) \\ &\leq +f\,;+f' \end{aligned}$$

∎

The concept of sum bears a close relation to the *Ab*-categorial notion of biproduct, but it is not in fact a limit construct. To make it so would require, because of the symmetry in bicontexts, making it a product and coproduct simultaneously, which seems impossibly restrictive. It is in fact a kind of hybrid of product and coproduct, as will be seen in the examples below where derive from coproducts and some from products in an underlying context. We shall take up this matter again in Chapter 4.

We turn now to the lifting of joins and sums by forgetful functors. In this respect, we want not only that they lift qua functors, but also that the the injections are suitably related.

Terminology.

Let $V : C \to C'$, be a functor, $+$ a prejoin on C, and $+'$ a prejoin on C'. We shall say that $+$ lifts $+'$ (via V) iff it does so qua functors, and V preserves all the injections, ie: every $V\eta_{x.i} = \eta'_{(Vx),i}$.

3.5.7 LEMMA. *Let $+$ lift $+'$ via exact V. Then we have, for $f \in C^I[x, y]$ and $g \in C[+x, +y]$*

(1) $Vf\_i = (Vf)\_i$

(2) $Vg| = Vg|$

PROOF:

(1)

$$(Vf)\_i = \eta'^- ; Vf ; \eta'$$

$$= V\eta^- ; Vf ; V\eta$$

$$= Vf\_i$$

(2)

$$Vg|i = V\eta_i ; Vg ; V\eta_i^-$$

$$= \eta' ; Vg ; \eta'^-$$

$$= Vg|i$$

∎

We can therefore write Vf_i and $Vg|$ unequivocally.

3.5.8 THEOREM. *Now let + lift +' via forgetful V. Then + is a (exact) join / (disjunctive) sum if +' is. Furthermore, if V is full and spanning the reverse is true (ie: +' is a ... if + is).*

PROOF:

(1) $+'$ is a join implies $+$ is: Let $f \in C^I$. Then $V{+}f = +'Vf \geq Vf\_i$. Thus $V{+}f \geq Vf\_i$, whence $+f \geq f\_i$ from bimonotonicity, so $+$ is a join.

(2) $+'$ is exact implies $+$ is: Use 3.3.13

(3) $+'$ is a sum implies $+$ is: We already know, by (1), that $+$ is a join. Let $f : +x \to +y$, and use 3.5.3(2) to calculate:

$$V{+}\,f| = +'\,Vf| \leq Vf$$

whence

$$+\,f| \leq f *$$

(4) $+'$ disjunctive implies $+$ is: Now let the f in (3) be a part on $+x$. Then Vf is a part on $V{+}x = +'Vx$, so we get equality in (*).

For the second half, V full and spanning means that any $f' \in C' = Vf\,\exists f \in C$.

(1) $+$ is a join implies $+'$ is: Let $Vf \in C'^I[Vx, Vy]$. Then $+'\,Vf = V{+}f \geq Vf\_i$.

(2) $+$ is exact implies $+'$ is: Given arrows in $C'^I[x', y']$ and $C'^I[y', z']$, choose x, y, z with $Vx = x'$, $Vy = y'$ and $Vz = z'$. Then the arrows are expressible as Vf, Vf' for some $f \in C^I[x, y]$, $f' \in C^I[y, z]$. We then have

$$\begin{aligned} +'\,Vf\,;Vf' &= +'V\ f\,;f' \\ &= V{+}\ f\,;f' \\ &= V(+f\,;+f') \\ &= V{+}f\,;V{+}f' \\ &= +'Vf\,;+'Vf' \end{aligned}$$

(3) $+$ is a sum implies $+'$ is: We already know that $+'$ is a join. Let $f' \in C'[+'x', +'y']$. Choose $x, y \in C^I$ with $Vx = x'$ and $Vy = y'$. Then $f' \in C'[V + x, V + y]$, so $f' = Vf$, $\exists f \in C$. We then get, using 3.5.3(2)

$$(\dagger) \qquad +'\,Vf| = V{+}\ f| \leq Vf$$

(4) $+$ disjunctive implies $+'$ is: Let the Vf in (3) now be a part on $+'x' = V{+}x$. Then $Vf \leq V{+}x$, so $f \leq +x$, ie: f is a part. So we get equality in ($\dagger$). ∎

One would expect that zero components contribute nothing to a sum. This is true in the sense that when all but one component in a sum is zero, the sum is isomorphic to the non-zero object.

3.5.9 PROPOSITION. *Let $i \in I$, and let $x \in C^I$ with all but x_i zero. Then*

$$\eta_{x,i} : x_i \cong +x$$

PROOF: We have only to show that $\eta_i^R = +x$. But $+x = \bigvee_{i' \in I} x\_i' = \bigvee_{i' \in I} \eta_{i'}^R = \eta_i^R$. ∎

Moreover, if the sum in 3.5.9 happens to be exact, sums with more zero components are "smaller" in the following sense. Let $x, y \in C^I$ be such that x is zero at every $i \in I$ where y is zero. Then there is an obvious injection from x to y, which $+$ (being exact) carries into an injection from $+x$ to $+y$.

It would be possible to specify a requirement that sums be commutative in the sense that the order of presentation of their arguments is immaterial. We shall not pursue this here because it is tedious to no real advantage.

Examples.

In **Set** —

(1) Disjoint union is an exact sum, with the obvious injections, because, for $r : +x \to +y$ in **Set**,

$$+ r| = \{\langle x_i, y_i\rangle \in r \mid i \geq 0\} \subseteq r.$$

It is disjunctive.

In $\mathbf{Set}_0$ —

(2) Cartesian product defined by, (for $r : x \to y \in \mathbf{Set}_0$):

$$+r = \{(x, y) \mid \forall i \in I\ (x_i = 0\ \&\ y_i = 0) \text{ or } (x_i\ r_i\ y_i)\} \setminus \{(0, 0)\}$$

with each $\eta_i = 0[i]$ is an exact join but not a sum. It is not disjunctive.

(3) Coalesced union, which is disjoint union but with all the 0s identified as a common 0, with the obvious injections and $+r = \bigcup_i r_i$ lifts (1) via *drop*. It is therefore an exact disjunctive sum.

In **Type2** —

(4) The standard coalesced sum construction yields a sum when

$$\eta_i = \langle (\cdot)\mathrm{in}{+}x, (\cdot)|x_i\rangle,$$

because for $f : +x \to +x'$,

$$f|i : x_i \mapsto (\text{if } fx_i : x_i' \text{ then } fx_i \text{ else } 0),$$

so $+ f| \leq f$. It is exact and disjunctive.

(5) Cartesian product is an exact sum, with

$$\eta_i = \langle x_i \mapsto \langle 0, \ldots, 0, x_i, 0, \ldots, 0\rangle, (\cdot)_i \rangle$$

because

$$f|i : x_i \mapsto f(0, \ldots, 0, x_i, 0, \ldots, 0)_i$$

so

$$(+ f|)x = \langle f(0, \ldots, 0, x_i, 0, \ldots, 0)_i \rangle_i \leq fx.$$

It is not disjunctive, for consider the function $\{0, 1, 2 \mapsto 0, 3 \mapsto 3\}$ on the type $\mathbb{2} = \mathbb{1} + \mathbb{1}$ (see Diagram 3.5). The only sum of parts $\leq \langle f, f \rangle$ is $0 + 0 = 0$. Notice that Cartesian product being a sum does not depend in any way on $\bigvee_{i \in I} f\_i$ being a *pointwise* sup.

(6) Standard separated sum (disjoint union with an extra 0) with the injections shown in Diagram 3.6 is an exact linear join, but not a sum because it is not monostrict. And it is not disjunctive because any part with a component which maps some $x > 0$ into 0 is not an image of parts.

(7) In all the above examples, the joins which are not sums are also not disjunctive. We now give an example of a disjunctive join that is not a sum. We take C to be the bicontext with one object whose bundle is the continuum $\mathbb{I} = [0, 1]$ under its natural ordering, with multiplication as composition and the identity as conjugation. 1 is the identity arrow, so every arrow is a part.

We now define the pre-join $+ : C^I \to C$ by

$$r \mapsto f(\bigvee_i r_i),$$

where $f : \mathbb{I} \to \mathbb{I}$ is any onto function greater than the identity on $\mathbb{I}$ (such as that shown in Diagram 3.7), with all injections equal to 1. Then every $r\_i = r_i$, and $r|i = r$, so $+$ is obviously a join. By the choice of f, it is not a sum, but because f is onto and every arrow is a part, if $r = fs$ then $r = +\langle s\rangle_i$, whence $+$ is disjunctive.

Furthermore, this join is not bimonotonic, whereas all the previous ones are.

We shall see more examples of joins and sums in Chapter 5.

3.6 Diagrams and Cones

For the subsequent sections of this chapter, let us fix a bicontext C and a net N. In this section we shall study the notion of a cone over a diagram in C with a view to obtaining a concept of limit in the next section.

Notation.

We shall use $m, n, ..$ for objects of N, and $a, b, ...$ for arrows of N. Also, diagrams over N will be denoted by various versions of $[\![\cdot]\!]$.

By an arrow between diagrams we shall mean an arrow in $\mathrm{Diag}(N, C)$. Thus, if $r : [\![\cdot]\!] \to [\![\cdot]\!]'$, we have

$$r_m \,;\, [\![a]\!]' \leq [\![a]\!] \,;\, r_n \quad \forall a : m \to n \in N$$

3.6.1 Definition. *A cone over* $[\![\ ]\!]$ *is an exact NT,* c *to some constant diagram* c_∞*. We denote this by* $[\![\cdot]\!] \rhd c$*.* c_∞ *is the* apex *of the cone.*

We shall refer to a diagram or cone as having some property of arrows to mean that each of its components has that property.

$\mathrm{Cone}(N, C)$ is the full sub of $\mathrm{Vert}(\mathrm{Diag}(N, C))$ whose objects are cones. An arrow between cones will be understood in this context. When $r : [\![\cdot]\!] \to [\![\cdot]\!]'$, $[\![\cdot]\!] \rhd c$, $[\![\cdot]\!]' \rhd$

d and $r_\infty : c_\infty \to d_\infty$ form a square (see Diagram 3.8), we have an arrow in Cone, which we shall write as $r_\infty : c \xrightarrow{r} d$.

Moreover, viewing r as a diagram in Hor(C), $\langle c, d\rangle$ is a cone over r with apex r_∞.

There are two instances of composition in Diag of interest here:

(1) Composition of an exact arrow between diagrams and a cone over the target diagram to obtain a co-apical cone over the source diagram:

$$r : [\![\cdot]\!] \xrightarrow{=} [\![\cdot]\!]' \rhd c \quad \text{gives} \quad [\![\cdot]\!] \rhd (r ; c)$$

(2) Composition of a cone and a C-arrow, treated as an exact arrow between constant diagrams, to give another cone over the same diagram:

$$[\![\cdot]\!] \rhd c \xrightarrow{f} x \quad \text{gives} \quad [\![\cdot]\!] \rhd (c ; f)$$

An immediate consequence of 3.1.21 is that an arrow between diagrams is symmetric if the diagrams are adjunctive.

Now assume that $[\![\cdot]\!]$ is total and consider, for given $n \in N$, the set

$$n_L \stackrel{\text{def}}{=} \{[\![a]\!]^L \mid a \text{ has source } n\}.$$

If a, a' both have source n, they have a unifier, say $\langle b, b'\rangle$. Let $a'' = a ; b = a' ; b'$. Then

$$\begin{aligned} [\![a'']\!]^L &= [\![a ; b]\!]^L \\ &= ([\![a]\!] ; [\![b]\!])^L \\ &= [\![a]\!] ; [\![b]\!]^L ; [\![a]\!]^- \\ &\geq [\![a]\!]^L \end{aligned}$$

Likewise, $[\![a'']\!]^L \geq [\![a']\!]^L$, so n_L is directed.

3.6.2 Definition. *For total diagram* $[\![\cdot]\!]$, *the* view of $[\![\cdot]\!]$ at n, $[\![n]\!]^*$, *is defined to be* $\bigvee n_L$.

Notice that $[\![n]\!]^* \geq 1$, with equality if $[\![\cdot]\!]$ is expanding.

3.6.3 PROPOSITION. *If cone c over $[\![\cdot]\!]$ is total, $c_m^L \geq [\![a]\!]^L$ for any $a : m \to n$ in N. Furthermore, if c is expanding, so is $[\![\cdot]\!]$, and it is determined by c, viz: $[\![a]\!] = c_m ; c_n^-$.*

PROOF:

$$\begin{aligned} c_m^L &= [\![a]\!] \,;\, c_n^L \,;\, [\![a]\!]^- \\ &\geq [\![a]\!]^L. \end{aligned}$$

If c is expanding this becomes

$$\begin{aligned} 1 &= [\![a]\!] \,;\, 1 \,;\, [\![a]\!]^- \\ &= [\![a]\!]^L \end{aligned}$$

Also

$$\begin{aligned} c_m \,;\, c_n^- &= [\![a]\!] \,;\, c_n \,;\, c_n^- \\ &= [\![a]\!] \end{aligned}$$

Thus, when $[\![\cdot]\!]$ is total, c is total iff every $c_n^L \geq [\![n]\!]^*$.

We can construct contexts to represent both the "bases" and the "apices" of cones. Define

$$apex : \mathrm{obj}(\mathrm{Cone}) \to \mathrm{obj}C : c \mapsto c_\infty$$

$$base : \mathrm{obj}(\mathrm{Cone}) \to \mathrm{obj}(\mathrm{Diag}) : ([\![\cdot]\!] \rhd c) \mapsto [\![\cdot]\!]$$

Then we can form the contexts

$$\mathrm{Apex}(N, C) = C \times apex$$

and

$$\mathrm{Base}(N, C) = \mathrm{Diag} \times base$$

If we have two cones and a C-arrow between their apices, we can construct an arrow between the diagrams which completes the apical arrow to an arrow between the cones.

3.6.4 DEFINITION. *Let* $f : c_\infty \to d_\infty$, *with* $[\![\cdot]\!] \triangleright c$, $[\![\cdot]\!]' \triangleright d$, *and* $[\![\cdot]\!]', d$ *both singular. Define* $\nabla f = c \,;\, f \,;\, d^-$.

3.6.5 THEOREM.

(1) $\nabla f : [\![\cdot]\!] \to [\![\cdot]\!]'$, *exact if* $[\![\cdot]\!]'$ *is contracting.*

(2) $f : c \xrightarrow{\nabla f} d$, *exact if* d *is contracting.*

PROOF:

(1)

$$\begin{aligned}
\nabla f_m \,;\, [\![a]\!]' &= c_m \,;\, f \,;\, d_m^- \,;\, [\![a]\!]' \quad (a : m \to n \in N)\\
&= [\![a]\!] \,;\, c_n \,;\, f \,;\, d_n^- \,;\, [\![a]\!]'^- \,;\, [\![a]\!]'\\
&\leq [\![a]\!] \,;\, c_n \,;\, f \,;\, d_n^- \quad \text{(with equality if } [\![a]\!]' \text{ is contracting)}\\
&= [\![a]\!] \,;\, \nabla f_n
\end{aligned}$$

(2) ∇f fills in $\langle ?, d \mid c, f\rangle \in C^N$, exactly if d is contracting. ∎

Now ∇ has the following obvious properties:

(1) $(\nabla f)^- = \nabla(f^-)$, so ∇f is symmetric if $[\![\cdot]\!], c$ are also singular.

(2) If $g : d_\infty \to e_\infty$ $\nabla(f \,;\, g) \geq \nabla f \,;\, \nabla g$ with equality if d is contracting.

(3) $\nabla c_\infty = c_\infty$ iff c is expanding.

(4) If $D \nearrow_C f$, we have

$$\begin{aligned}
\nabla f_n &= \bigvee_{f' \in D} (c_n \,;\, f' \,;\, d_n^-)\\
&= \bigvee_{f' \in D} \nabla f'_n
\end{aligned}$$

so

$$\nabla f = \bigvee_{f' \in D} \nabla f'$$

Thus ∇ is an upper static functor : $\mathrm{Apex}_{\mathrm{Inj}} \to \mathrm{Cone}$, pivot-exact on contracting cones, where

$$\mathrm{Apex}_{\mathrm{Inj}} = \mathrm{Apex} \mid \{\text{injective cones over singular diagrams}\}.$$

But an injective cone which is also contracting is an iso, so the exactness condition can be rephrased as "pivot-exact on iso cones". And 3.6.5(2) says that

$$apex\ \nabla = \text{identity}.$$

The action of functors on diagrams and cones.

We now look at how these various entities are transformed by functors. We shall apply the results of 3.3 separately to a K-functor F $(K < C)$ and a thread functor T, both within C.

In the first case, assume that all diagrams and cones are in K. Then every diagram satisfies K (ie: $K \times C \cup C \times K$), as does every arrow between diagrams and every arrow between cones. Thus:

3.6.6 PROPOSITION.

(1) *$F[\![\cdot]\!]$ is a diagram, in K, over N.*

(2) *If $r : [\![\cdot]\!] \to [\![\cdot]\!]'$, then $Fr : F[\![\cdot]\!] \to F[\![\cdot]\!]'$, exact if r is. In particular, if c is a cone, so is Fc.*

(3) *If $r_\infty : c \xrightarrow{r} d$, then $Fr_\infty : Fc \xrightarrow{Fr} Fd$. It is exact if r_∞ is.*

In the second case, assume that all diagrams satisfy fit. Notice that a cone satisfies fit when every $[\![a : m \to n]\!], c_n$ fit.

3.6.7 PROPOSITION.

(1) *$T[\![\cdot]\!]$ is a smooth diagram.*

(2) *If $r : [\![\cdot]\!] \to [\![\cdot]\!]'$ satisfies fit, then $Tr : T[\![\cdot]\!] \to T[\![\cdot]\!]'$, exact if r is. Again, if c is a cone, so is Tc.*

(3) *If $[\![\cdot]\!] \rhd c$ and c satisfies fit, then $T[\![\cdot]\!] \rhd Tc$.*

(4) *If $r_\infty : c \xrightarrow{r} d$, and r, d and c, r_∞ fit, then $Tr_\infty : Tc \xrightarrow{Tr} Td$.*

3.7 LIMITS

In this section we see how the concept of limit arises in bicontexts. It is most easily defined as a special case of a more general notion, which we shall investigate first. We assume here that all diagrams (again over N) are adjunctions. It is then immediate that every diagram satisfies fit and every arrow between diagrams is symmetric.

It is interesting that we seem to need to make this assumption in order to get a notion of limit. In [1] Gunter shows that (direct) limits exist in the context Adj(Type2) following work by Niño [3] on adjunctions. The implication is that if a limit construction in some context is really taking place in a wider bicontext, the adjunction constraint on the diagram is going to be required.

3.7.1 DEFINITION. *Cone c over $\llbracket\cdot\rrbracket$ is tight iff every $c_n^L = \llbracket n\rrbracket^*$.*

By 3.6.3, c is tight iff it is total and every $c_n^L \leq \llbracket n\rrbracket^*$.

Henceforth in this section, cones will be *total* unless otherwise stated.

3.7.2 LEMMA. *For every $a : m \to n$, $c_m^- ; r_m ; d_m \leq c_n^- ; r_n ; d_n$*

PROOF:

$$\begin{aligned} c_m^- ; r_m ; d_m &= c_n^- ; \llbracket a\rrbracket^- ; r_m ; \llbracket a\rrbracket' ; d_n \\ &\leq c_n^- ; \llbracket a\rrbracket^R ; r_n ; d_n \\ &\leq c_n^- ; r_n ; d_n \end{aligned}$$

∎

Thus, since N is a net, the set $\{c_n^- ; r_n ; d_n \mid n \in N\}$ is directed. So we can make:

3.7.3 DEFINITION. *The* modulus *of c, r, d is*

$$|c, r, d| = \bigvee_{n\in N} c_n^- ; r_n ; d_n : c_\infty \to d_\infty$$

In case $\llbracket\cdot\rrbracket = \llbracket\cdot\rrbracket'$, we write $|c, d|$ for $|c, \llbracket\cdot\rrbracket, d|$ and $|c|$ for $|c, c|$. If $|c| = 1$, c is a unit *cone.*

Notice that if we treat r as a diagram in $\mathrm{Hor}(C)$, $\langle c, d\rangle$ is a cone over r whose modulus is $\langle |c|, |d|\rangle$.

We first list without proof some basic properties of modulus.

3.7.4 PROPOSITION.

(1) $|c, r, d|$ *is continuous in all three arguments, ie: qua function*

$$|\cdot,\cdot,\cdot| : \mathrm{Diag}[[\![\cdot]\!], c_\infty] \times \mathrm{Diag}[[\![\cdot]\!], [\![\cdot]\!]'] \times \mathrm{Diag}[[\![\cdot]\!]', d_\infty] \to C[c_\infty, d_\infty]$$

(2) $|c| \leq 1$ *iff* c *is singular.*

(3) $|c, r, d|^{-} = |d, r^{-}, c|$

(4) *If* $r : [\![\cdot]\!] \xrightarrow{=} [\![\cdot]\!]'$ *, then* $|c, r, d| = |c, (r\,;d)|$ *(the exactness here is to ensure that* $r\,; d$ *is a cone).*

The next proposition describes the behaviour of the composite of moduli with respect to the modulus over a composite arrow.

3.7.5 PROPOSITION. $|c, r, d|\,;|d, s, e| \geq |c, (r\,;s), e|$*; with equality if* d *is tight.*

PROOF:

$$\begin{aligned}
|c, r, d|\,;|d, s, e| &= \bigvee_n (c_n^- \,; r_n \,; d_n^L \,; s_n \,; e_n)\\
&\geq \bigvee_n (c_n^- \,; r_n \,; s_n \,; e_n)\\
&= |c, (r\,;s), e|.
\end{aligned}$$

If d is tight, we can continue from the first line with:

$$\begin{aligned}
&= \bigvee_{a:n\to n'} (c_n^- \,; r_n \,; [\![a]\!]''^{L} \,; s_n \,; e_n)\\
&\leq \bigvee_{a:n\to n'} (c_n^- \,; [\![a]\!] \,; r'_n \,; s'_n \,; [\![a]\!]''^{-} \,; e_n)\\
&= \bigvee_{a:n\to n'} (c'^{-}_n \,; [\![a]\!]^R \,; r'_n \,; s'_n \,; [\![a]\!]''^R \,; e'_n)\\
&\leq \bigvee_{n'} (c'^{-}_n \,; r'_n \,; s'_n \,; e'_n)\\
&= |c, (r\,;s), e|
\end{aligned}$$

∎

3.7.6 COROLLARY. *In particular,* $|c,d|\,;|d,c| \geq |c|$, *and if* c *and* d *are both tight unit cones,* $|c,d| : c_\infty \cong d_\infty$.

3.7.7 THEOREM. $|c,r,d| : c \xrightarrow{r} d$, *exact if* c *is tight and* r *is exact.*

PROOF: The first part is immediate because, c being total, $c^- \,; r \,; d$ fills in the right side of $\langle r, d \mid c, ?\rangle$ in C^N. For the exactness we have

$$\begin{aligned}
c_m \,;|c,r,d| &= c_m \,; \bigvee_n (c_n^- \,; r_n \,; d_n) \\
&= \bigvee_{a:m\to n} (c_m \,; c_n^- \,; r_n \,; d_n) \quad \text{because } N \text{ is a net} \\
&= \bigvee_{a:m\to n} ([\![a]\!] \,; c_n^L \,; r_n \,; d_n) \\
&= \bigvee_{a:m\to n,\, b:n\to n'} ([\![a]\!] \,; [\![b]\!]^L \,; r_n \,; d_n) \\
&\leq \bigvee_{a,b} ([\![a \,; b]\!] \,; r'_n \,; [\![b]\!]'^L \,; d_n) \quad \text{because r is symmetric} \\
&= \bigvee_{a,b} ([\![a \,; b]\!] \,; r'_n \,; [\![b]\!]'^R \,; d'_n) \\
&\leq \bigvee_{a,b} ([\![a \,; b]\!] \,; r'_n \,; d'_n) \\
&= \bigvee_{a,b} (r_m \,; [\![a \,; b]\!]' \,; d'_n) \quad \text{because r is exact} \\
&= \bigvee_{a,b} (r_m \,; d_m) \\
&= r_m \,; d_m \quad \text{since m is fixed}
\end{aligned}$$

∎

Some of the behaviour of modulus can be encapsulated in the statement that it is a lower static functor from Base_1 to Cone, where

$$\mathrm{Base}_1 = \mathrm{Base} \mid \{\text{total unit cones over adjunctive digrams}\},$$

which is pivot-exact on tight cones. And *base* $|\cdot| =$ identity.

3.7.8 THEOREM. *If* $r_\infty : c \xrightarrow{r} d$, *then* $|c,r,d| \leq |c|\,; r_\infty$, *with equality if* r_∞ *is exact.*

PROOF:

For any n,

$$r_n \,;\, d_n \leq c_n \,;\, r_\infty$$

Thus

$$c_n^- \,;\, r_n \,;\, d_n \leq c_n^- \,;\, c_n \,;\, r_\infty$$

whence

$$|c, r, d| \leq |c| \,;\, r_\infty$$

with equality all through if r_∞ is exact. ∎

The next Corollary gives the essential properties of the modulus.

3.7.9 COROLLARY. *Let c be a unit cone. Then*

(1) $|c, r, d|$ *is least amongst arrows from c to d over r.*

(2) *There is at most one exact arrow from c to d over r, which is $|c, r, d|$.*

(3) *If c is tight and r is exact, $|c, r, d|$ is the* unique *exact arrow from c to d over r.*

Furthermore, for tight unit cones (over the same diagram), 3.7.6 ensures that their modulus is a unique iso between them.

This corollary tells us that tight unit cones are simultaneously limits and colimits in the underlying category. However, we shall call them simply limits, and reserve the term colimit for a special case. We therefore make the following definition:

3.7.10 DEFINITION.

(1) *A tight unit cone over an adjunctive diagram is* a limit.

(2) *An injective limit is* a colimit.

(3) *An iso colimit is* degenerate.

Notice that, by 3.7.4(2), a limit is singular, and therefore an adjunction. And by 3.6.3, the base of a colimit is injective and uniquely determined by the colimit. In this case each $c_n^L = [\![n]\!]^* = 1$, so c is necessarily tight. Thus a colimit is simply an injective unit cone. Moreover, if a diagram has a colimit, all its limits are colimits.

Hence we have that, qua functors from Base_1 to Cone, modulus is pivot-exact on limits, and ∇ is defined on colimits and pivot-exact on degenerate ones.

It is also worth remarking that if we treat r as a diagram in $\mathrm{Hor}(C)$, we find that the cone $\langle c, d\rangle$ with apex $|c, r, d|$ in Hor is total and tight (injective), when both c and d are, and therefore is a (co)limit when c and d are.

3.7.11 PROPOSITION. *If cone c over $[\![\cdot]\!]$ is a limit and $r : c_\infty \cong x$, then $c\,;r$ is a limit.*

PROOF:

$$\begin{aligned}(c\,;r)_n^L &= c_n\,; r^L\,; c_n^-\\ &= c_n^L\\ &= [\![n]\!]^*\end{aligned}$$

so $c\,;r$ is tight.

$$\begin{aligned}|c\,;r| &= sup_n(r^-\,; c_n^R\,; r)\\ &= r^-\,; sup_n(c_n^R)\,; r\\ &= r^-\,; |c|\,; r\\ &= r^R\\ &= 1\end{aligned}$$

Thus $c\,;r$ is also unit.

∎

The action of functors on limits.

We now look at how functors affect moduli and limits. Let $K < C$ and assume that all diagrams and cones are in K. Let F be a functor that is K-end-exact. This implies that every $F[\![\cdot]\!]$ is adjunctive and every Fc is total.

3.7.12 LEMMA.

(1) $\forall n \in N,\ F([\![n]\!]^*) \leq (F[\![n]\!])^*$

(2) *If c is tight, so is Fc.*

PROOF:

(1)

$$\begin{aligned} F([\![n]\!]^*) &= \bigvee_{a:n \to n'} (F[\![a]\!]^L) \\ &\leq \bigvee_{b:Fn \to n'} [\![b]\!]^L \\ &= (F[\![n]\!])^* \end{aligned}$$

(2)

$$\begin{aligned} Fc_n^L &\leq F([\![n]\!]^*) \\ &\leq (F[\![n]\!])^* \quad \text{by (1)} \end{aligned}$$

∎

We consider the two particular cases of a K-functor and a thread functor, both of which are indeed K-end-exact. Observe that K-end-exactness preserves injections in K.

So let F be a K-functor.

3.7.13 PROPOSITION. $|Fc, Fr, Fd| = F|c, r, d|$

PROOF:

$$\begin{aligned} |Fc, Fr, Fd| &= \bigvee_n (Fc_n^- \,;\, Fr_n \,;\, Fd_n) \\ &= F \bigvee_n (c_n^- \,;\, r_n \,;\, d_n) \\ &= F|c, r, d| \end{aligned}$$

∎

3.7.14 COROLLARY. *If c is a (co)limit in K, so is Fc.*

PROOF: By 3.7.12, Fc is tight. Also $|Fc| = F|c| = 1$ and K-end-exactness guarantees that Fc is injective if c is. ∎

Now let T be a thread functor.

3.7.15 PROPOSITION. *$|Tc, Tr, Td| \geq T|c, r, d|$ with equality if c^-, r and r, d fit.*

PROOF:

$$\begin{aligned} |Tc, Tr, Td| &= \bigvee_n (Tc_n^- \,;\, Tr_n \,;\, Td_n) \\ &\geq T \bigvee_n (c_n^- \,;\, r_n \,;\, d_n) \quad (= \text{if } c^-, r \text{ and } r, d \text{ fit}) \\ &= T|c, r, d| \end{aligned}$$

∎

3.7.16 COROLLARY. *If c is a (co)limit, so is Tc.*

PROOF: By 3.7.12, Tc is tight. Also $|c| = |c, [\![\cdot]\!], c|$. Since c is adjunctive, $c^-, [\![\cdot]\!]$ and $[\![\cdot]\!], c$ fit. Thus $|Tc| = T|c| = 1$. Again, K-end-exactness guarantees that Tc is injective if c is. ∎

Finally in this section, we look at the possibility of extending the domain of definition of a functor to include certain limit objects (ie: the apices of limit cones); Chapter 5 will present an application of the construction.

Let $J < K \lhd^{=} C$ full such that every $x \in \mathrm{obj}C$ is the apex of a colimit over some N-diagram in J, and that every such diagram has a colimit. If F is a J-functor from K to C, we show how to extend it to act upon the whole of C.

Given $[\![\cdot]\!] \in \mathrm{Diag}(N, J)$, we know that $F[\![\cdot]\!]$ is also one. Now define $([\![\cdot]\!] \rhd c,\ [\![\cdot]\!]' \rhd d$ and $f : c_\infty \to d_\infty)$

$$\bar{F} : \mathrm{Apex}|\mathrm{colimits} \to \mathrm{Cone}|\mathrm{colimits}$$

by

$$\bar{F}c = \text{ some (necessarily co-)limit over } F[\![\cdot]\!] \quad \text{AoC}$$

$$\bar{F}f = |\bar{F}c, F\nabla f, \bar{F}d|$$

This is well-defined because $\nabla f \in K$ since K is full. In fact, $\bar{F}$ is the composite functor $|\cdot|\ F\ \mathit{base}\ \nabla$.

Since F is upper and modulus is pivot-exact on colimits, $\bar{F}$ is upper. We can now extend F to the whole of C by

$$F(f : x \to y) = \mathit{apex}\ \bar{F}(f : c_\infty \to d_\infty)$$

where $x = c_\infty, y = d_\infty$.

3.8 The Limit Functor, F^ω

In this section, F is a K-functor, with $K < C$. We also take N to be the category $\{\langle m, n\rangle \mid m \leq n \in \omega\}$, with objects $\langle n, n\rangle$ (abbreviated to n), $\langle m, n\rangle : m \to n$, and $\langle m, n\rangle ; \langle n, p\rangle = \langle m, p\rangle$. It is easy to see that this is a net.

3.8.1 Definition. *Adjunction $k : x \to Fx \in K$ is an F-seed.*

Note that $0 : x \to Fx$ is an F-seed iff x is a zero object. We now use F to generate a diagram from an F-seed in a standard manner.

The diagram $[\![\cdot]\!]_{k,F} : N \to K$ is defined by (drop the subscript(s) when there is no confusion, and write $[\![m, n]\!]$ for $[\![\langle m, n\rangle]\!]$)

$$[\![n]\!] = F^n x$$

$$[\![m, n]\!] = F^m k \,;\, \cdots ;\, F^{n-1} k \quad \text{for } m < n$$

We shall call such diagrams F-diagrams.

3.8.2 Proposition. *$[\![\ ]\!]_{k,F}$ is an adjunctive diagram, with $F^j [\![m, n]\!] = [\![m+j, n+j]\!]$.*

Now suppose that $[\![\cdot]\!] \triangleright i$ is a limit in K, whence $F[\![\cdot]\!] \triangleright Fi$ is also. Such an i we shall call an F-limit.

3.8.3 DEFINITION. *Define i^+ by $i^+_n = i_{n+1}$.*

3.8.4 PROPOSITION. *i^+ is a limit over $F[\![\cdot]\!]$, with $i^+_\infty = i_\infty$.*

PROOF: Obviously $|i^+| = |i| = 1$.

$$\begin{aligned}(i^+_n)^L &= i^L_{n+1} \\ &= [\![n+1]\!]^* \\ &= \bigvee_{n' \geq n} [\![n+1, n'+1]\!]^L \\ &= \bigvee_{n' \geq n} F[\![n, n']\!]^L \\ &= (F[\![n]\!])^*\end{aligned}$$

∎

There is therefore a unique iso between i_∞ and Fi_∞ over $F[\![\cdot]\!]$, namely $|i^+, Fi|$. We shall refer to this modulus as $\ddot{u}$. Now define:

3.8.5 DEFINITION. *$next = \{next_n \mid n \geq 0\}$, where $next_n = F^n k = [\![n, n+1]\!]$.*

It is obvious that $next : [\![\cdot]\!] \xrightarrow{=} F[\![\cdot]\!]$. We can use it to describe $\ddot{u}$ as a modulus involving both i and Fi.

3.8.6 THEOREM. $\ddot{u} = |i, next, Fi|$

PROOF:

$$\begin{aligned}\ddot{u} &= \bigvee_n (i^-_{n+1} \,;\, Fi_n) \\ &= \bigvee_n (i^-_{n+1} \,;\, F(next_n \,;\, i_{n+1})) \\ &= \bigvee_n (i^-_{n+1} \,;\, next_{n+1} \,;\, Fi_{n+1})\end{aligned}$$

∎

3.8.7 COROLLARY. *$next\,; Fi$ is a limit (over $[\![\cdot]\!]$).*

PROOF: Because $next$ is exact, $i\,; \ddot{u} = next\,; Fi$ and $\ddot{u}$ is an iso. ∎

Now let $k' : x' \to Fx'$ be another F-seed with F-limit i', and suppose there is $r_0 : x \to x'$ such that $\langle r_0, Fr_0 \rangle : k \to k'$ in $\mathrm{Vert}(C)$. We shall call r_0 an *F-seed* *arrow from k to k'*, and write $r_0 : k \xrightarrow{0} k'$. If we have equality, r_0 is *exact*, and we write $r_0 : k \xrightarrow{0=} k'$. Notice that, if $k : x \to Fx$ is an F-seed, so is Fk and k itself is an F-seed arrow from k to Fk. Given this situation, we make:

3.8.8 DEFINITION. *$r_n = F^n r_0$ and $F^* r_0 = \{r_n \mid n \geq 0\}$.*

3.8.9 THEOREM. *$F^* r_0 : [\![\cdot]\!]_{k,F} \to [\![\cdot]\!]_{k',F}$, with exactness if r_0 is exact.*

PROOF: Writing $[\![\cdot]\!]$ and $[\![\cdot]\!]'$ for the two diagrams and r for $F^* r_0$, we have

$$\begin{aligned} r_m \,;\, [\![m,n]\!]' &= F^m r_0 \,;\, F^m k' \,;\, [\![m+1,n]\!]' \\ &= F^m(r_0 \,;\, k') \,;\, [\![m+1,n]\!]' \\ &\leq F^m(k \,;\, Fr_0) \,;\, [\![m+1,n]\!]' \quad (= \text{if } r_0 \text{ exact}) \\ &= F^m k \,;\, F^m + 1 r_0 \,;\, [\![m+1,n]\!]' \\ &= [\![m,m+1]\!] \,;\, r_m + 1 \,;\, [\![m+1,n]\!]' \end{aligned}$$

A simple induction on $(n - m)$ establishes the result. ∎

The properties of squares tell us that the F-seed arrows from k to k', are a subtype of Vert$[k, k']$. And since F is upper, the composite of F-seed arrows is also one. Thus the F-seed arrows form a bisub Seed$_F \lhd^{=}$ Vert (symmetric because the seeds are adjunctions). Clearly the composite of exact F-seeds is exact if either is in K. Furthermore, we have

3.8.10 LEMMA. *Let $r_0 : k \xrightarrow{0} k'$, $s_0 : k' \xrightarrow{0} k''$ be F-seeds. then $F^* r_0 \in K$ if $r_0 \in K$, and $F^*(r_0 \,;\, s_0) \geq F^* r_0 \,;\, F^* s_0$ with $=$ if either r_0 or s_0 is in K.*

PROOF: If $F^n(r_0 ; s_0) \geq F^n r_0 ; F^n s_0$, then $F^{n+1}(r_0 ; s_0) \geq F^{n+1} r_0 ; F^{n+1} s_0$ because F is upper. Use induction on n. If r_0 (say) is in K, so is every $F^n r_0$, whence these inequalities become equalities, and $F^* r_0 \in K$. ∎

Clearly F^* is continuous and symmetric, so we have

$$F^* \text{ is an upper functor } : \text{Seed}_F \to \text{Diag}|\{F\text{-diagrams}\}$$

In fact, we can say that F^* is a K-functor, with the obvious understanding of "$K <$ Seed$_F$".

3.8.11 DEFINITION. $F^{\omega}_{i,i'} r_0 = |i, r, i'|$. *Omit the subscripts when i, i' are understood.*

3.8.12 PROPOSITION. $F^{\omega} r_0 : i \xrightarrow{r} i'$ *, with exactness if r_0 is exact and in K if r_0 is.*

Using the previous Lemma, we see, since modulus is pivot exact on limits, that F^{ω} becomes a K-functor

$$f^{\omega} : \mathrm{Seed} \times start \to \mathrm{Cone}|\{F\text{-limits}\}$$

where

$$start : F\text{-limits} \to F\text{-seeds} : (\llbracket \cdot \rrbracket \triangleright i) \mapsto \llbracket 0, 1 \rrbracket$$

If we now assume that every F-diagram has a limit, and if we choose one for each diagram (AoC) to furnish the object-function, we can make F^{ω} into a K-functor over Seed itself:

$$F^{\omega} : \mathrm{Seed}_F \to \mathrm{Cone}|\{F\text{-limits}\}$$

In this way F^{ω} becomes a left inverse for the functor

$$seed \stackrel{\text{def}}{=} (\cdot)_0\ base$$

Thus, with a slight abuse of terminology, we can say that F^{ω} is also a K-functor. And the iso $\ddot{u}$ above is just $F^{\omega} next_0 = F^{\omega} k$.

3.9 SOLUTIONS OF FUNCTORS

The kind of limit construction just described was originally introduced by Scott as a way of "solving F", ie: finding an type x with $x \cong Fx$. We now look at how such solutions can appear as parts of objects in a bicontext.

First we define a way of treating parts as a kind of "subobject" and of viewing an interior $: x \to y$ as injecting $p \le x$ into $q \le y$. Henceforth in this section, C will be some fixed interior bicontext. Also, let $K <^{=} C$ and $F : C \to C$ be a K-functor.

Relative Injections.

3.9.1 DEFINITION. *Let $f : x \to y$ with $p \le x, q \le y$. then f is a* relative injection, *when $p \le f\,;q\,;f^{-}$. Write $f : p \Rrightarrow q$. When $f^{-} : q \Rrightarrow p$ as well, f is a* relative iso, *written $f : p \cong q$.*

3.9.2 LEMMA. *If $f : p \Rrightarrow q$, then $q \ge f^{-}\,;p\,;f$*

PROOF:

$$\begin{aligned} f^{-}\,;p\,;f &\le f^{-}\,;f\,;q\,;f^{-}\,;f \\ &\le q \end{aligned}$$

∎

3.9.3 COROLLARY. *If $f : p \cong q$, then $p = f\,;q\,;f^{-}$ and $q = f^{-}\,;p\,;f$.*

We list without proof some basic properties of relative injections, which justify the terminology.

Let $f : x \to y,\ g : y \to z$ and let $p \le x, q \le y, r \le z$.

- If $f : p \Rrightarrow q$ and $p' \le p, q \le q', f \le f'$, then $f' : p' \Rrightarrow q'$
- $f : x \Rrightarrow y$ iff f is an injection
- $1 : p \Rrightarrow p'$ iff $p \le p'$
- If $f : p \Rrightarrow q$ and $g : q \Rrightarrow r$, then $f\,;g : p \Rrightarrow r$
- $f : p \Rrightarrow q$ then $f : p^{-} \Rrightarrow q^{-}$

Now let p, q, r all be in $\mathrm{P}x$

- If $p : q \Rrightarrow r$, then $q \le \{p, p^{-}, r\}$
- If $p : q \cong r$, then $q = r \le \{p, p^{-}\}$

3.9.4 THEOREM. *For $p \leq x$, the following are equivalent:*

(1) p *is strong*

(2) $p = p^{-} = p \,;\, p$

(3) $p : p \cong p$

(4) $p : p \cong\!\!\rangle\; x$

PROOF: We already know from 3.2 that (1) and (2) are equivalent.

(2) implies (3): $p : p \cong p \;\Leftrightarrow\; p \leq p \,;\, p \,;\, p^{-} \;\; \& \;\; p \leq p^{-} \,;\, p \,;\, p$

(3) implies (4): Immediate from the list above.

(4) implies (2): $p : p \cong\!\!\rangle\; x \;\Rightarrow\; p \leq p \,;\, p^{-} \leq p^{-} \;\Rightarrow\; p = p^{-} = p \,;\, p$ ∎

3.9.5 DEFINITION. *Let $f : x \to Fx$ and $p \leq x$ such that $f : p \cong Fp$. Then p is a* solution of F in medium f. *Omitting the medium signifies "in some medium".*

It is immediate that x itself is a solution iff the medium is an iso.

We now want to consider how a part of x may generate a solution. To this end, we make the following definitions. Let $f : x \to Fx$, $g : y \to Fy$. Then $\binom{f}{g}_F$ is the recursor $\lambda r.(f \,;\, Fr \,;\, g^{-})$ on $C[x, y]$, and $f /_{a,F}\, g \stackrel{\text{def}}{=} \operatorname{fix}_a \binom{f}{g}_F$ (we shall omit the F when it is unambiguous). Then we say that a *generates* $r : x \to y$ (via f, g) when $a \leq r \leq f /_a\, g$. r is *inductive* when 0 generates it.

Thus, in particular, we can talk of a "solution generated by a", an "inductive solution", etc.

Notice that $\binom{f}{f}$ takes Px to Px, and that $a \leq 1 \;\Rightarrow\; f /_a\, f \leq 1$. In this case $f : f /_a\, f \cong\!\!\rangle\; F(f /_a\, f)$ and $: a \cong\!\!\rangle\; Fa$.

Now if $p \leq x$ is a solution, it is a fixed-point of $\binom{f}{f}$, but does the converse hold? A solution must also satisfy $Fp = f^{-} \,;\, p \,;\, f$. This will follow if f is a projection; for then RHS $= f^{R} \,;\, Fp \,;\, f^{R} = Fp$. Thus

3.9.6 PROPOSITION. *The solutions in projection f are precisely the fixed-points of $\binom{f}{f}$.*

In particular, $f\,/\,f$ is the *least* solution and the *only* inductive one.

Henceforth, a medium will always be a projection in K.

How can solutions in different media be compared? The obvious starting point is to examine the situation where $f : x \to Fx$, $g : y \to Fy$ and $h : z \to Fz$ are media with $r : x \to y$ and $s : y \to z$ (see Diagram 3.9). To do this, we must look more closely at the behaviour of $\binom{\cdot}{\cdot}$ and $/$. In the situation just described we have

$$\begin{aligned} \left(\binom{f}{g} r\right)^- &= (f\,;Fr\,;g^-)^- \\ &= g\,;Fr^-\,;f^- \\ &= \binom{g}{f}(r^-) \end{aligned}$$

so, if a is a seed for $\binom{f}{g}$, induction on n gives

$$\left(\binom{f}{g}^n a\right)^- = \binom{g}{f}^n (a^-),$$

for every $n \geq 0$, whence

3.9.7 PROPOSITION. $(f\,/_a\, g)^- = g\,/_a\, f$

3.9.8 LEMMA. $\binom{f}{h}(r\,;s) \geq \binom{f}{g}r\,;\binom{g}{h}s$ *with* $=$ *if r or s is in K.*

PROOF:

$$\begin{aligned} \binom{f}{h}(r\,;s) &= f\,;F(r\,;s)\,;h^- \\ &\geq f\,;Fr\,;Fs\,;h^- \quad (= \text{if } r \text{ or } s \text{ is in } K) \\ &= f\,;Fr\,;g\,;g^-\,;Fs\,;h^- \\ &= \binom{f}{g}r\,;\binom{g}{h}s \end{aligned}$$

∎

We can now prove

3.9.9 THEOREM. *Let $a : x \to y, b : y \to z$ be seeds for $\binom{f}{g}, \binom{g}{h}$ respectively. Then*

$$f \mathbin{/_{a}} g \,;\, g \mathbin{/_{b}} h \leq f \mathbin{/_{a;b}} h$$

with equality if either a or b is in K.

PROOF: By the previous Lemma, $r \,;\, s \leq t \;\Rightarrow\; \binom{f}{g}r \,;\, \binom{g}{h}s \leq \binom{f}{h}t$, so induction on n gives

$$\binom{f}{g}^{n} a \,;\, \binom{g}{h}^{n} b \leq \binom{f}{h}^{n} (a \,;\, b)$$

for every $n \geq 0$, whence

$$f \mathbin{/_{a}} g \,;\, g \mathbin{/_{b}} h \leq f \mathbin{/_{a;b}} h.$$

Moreover, under the extra conditions either every $\binom{f}{g}^{n}$ or every $\binom{g}{h}^{n}$ is in K, so every

$$\binom{f}{g}^{n} a \,;\, \binom{g}{h}^{n} b = \binom{f}{h}^{n} (a \,;\, b),$$

and equality holds in the result. ∎

Now suppose that $a : x \to x$ and $b : y \to y$ are seeds for $\binom{f}{f}$ and $\binom{g}{g}$ respectively, and that $r \in K$ is a seed for $\binom{f}{g}$. Then

$$(f \mathbin{/_{r}} g) \,;\, (g \mathbin{/_{b}} g) \,;\, (f \mathbin{/_{r}} g)^{-} = (f \mathbin{/_{r}} g) \,;\, (g \mathbin{/_{b}} g) \,;\, (g \mathbin{/_{r^{-}}} f)^{-}$$

$$= f \mathbin{/_{r;b;r^{-}}} f \quad \text{by the previous theorem, since } r, r^{-} \in K.$$

So if a, b are parts and $r : a \cong\rangle b$, we have

$$a \leq r \,;\, b \,;\, r^{-} \;\Rightarrow\; f \mathbin{/_{a}} f \leq f \mathbin{/_{r;b;r^{-}}} f = (f \mathbin{/_{r}} g) \,;\, (g \mathbin{/_{b}} g) \,;\, (f \mathbin{/_{r}} g)^{-}$$

$$\Rightarrow\; f \mathbin{/_{r}} g : f \mathbin{/_{a}} f \cong\rangle g \mathbin{/_{b}} g$$

In particular, if $r : a \cong b$, then $f \mathbin{/_{r}} g : f \mathbin{/_{a}} f \cong g \mathbin{/_{b}} g$, so isomorphic seeds produce isomorphic solutions in whatever medium. And since $0 : 0 \cong\rangle b$ and $0 : 0 \cong 0$, every inductive solution injects into any solution, and all inductive solutions are isomorphic. Recalling Section 3.8, if $k : x \to Fx$ is an F-seed, we saw that $F^{\omega}k : F^{\omega}x \cong F(F^{\omega}x)$, so $F^{\omega}x$ is a solution of F. In particular, we have

3.9.10 THEOREM. $F^{\omega}0$ *is the inductive solution of* F.

3.10 DOUBLETS

In this section we introduce a means of relating together two bicontexts in such a way that they have many properties in common. The concept bears a similarity to adjoints in conventional category theory, but the ordering replaces the natural transformations of the latter. An application of the correspondence will appear in Chapter 6.

3.10.1 DEFINITION. *A* doublet *is a pair* H, L *of bicontexts (the* high *and* low *halves) and thread functors* $down : H \leftrightarrows L : up$ *such that, for any* $h \in H,\ l \in L$

(1) $up\ down\ h \leq h$

(2) $down\ up\ l \geq l$

(3) $down\ 0 = 0$

It is coincident *when* H, L *are coincident and* $up, down$ *are static. The* stable *arrows of* H, L *are those with equality in (1) and (2).*

Note that (1) and (2) imply that $up\ down\ h \parallel h$ and $down\ up\ l \parallel l$.

3.10.2 DEFINITION. $h \in H,\ l \in L$ match *when* $down\ h = l$ *and* $up\ l = h$.

The following basic properties of a doublet are immediate.

- $down\ up\ down = down$
- $up\ down\ up = up$
- $up\ 0 = 0$ (so every 0 is stable).
- up, $down$ constitute a bijection of objects, ie: all objects are stable.
- Matching arrows are stable. Conversely, stable h matches $down\ h$ and stable l matches $up\ l$.

The next proposition tells us that doublets restrict to doublets on suitable subs.

3.10.3 PROPOSITION. *If $H' \trianglelefteq^{=} H$, $L' \trianglelefteq^{=} L$, down takes H' to L' and up takes L' to H', then down : $H' \leftrightarrows L'$: up is also a doublet, coincident if the original is. In particular, down :* $\mathrm{Part}(H) \leftrightarrows \mathrm{Part}(L)$ *: up is a doublet.*

The halves of a doublet can have common properties by dint of various things, not just arrows, being made to *match*. Of particular importance in this regard are matching translators, recursors, functors, limits and solutions.

Matching Translators, Recursors and Functors.

3.10.4 DEFINITIÒN. *Let T_H, T_L be translators in H, L respectively. They* match *when, for appropriate $l \in L$, $h \in H$ the following hold:*

(1) *$up\ T_L l \leq T_H up\ l$*

(2) *$down\ T_H h \geq T_L down\ h$*

with equality when l in (1), or h in (2) is stable.

Thus the bundles that T_H, T_L act between must correspond under the doublet. We shall refer to a stability-preserving translator as *stable*.

Without labouring the point too much, we may allow the Ts to be polyadic, in which case *up, down* will, as appropriate, be taken to mean $up^I, down^I$.

The two conditions in this definition are actually equivalent, although the associated equalities are not. The next proposition gives the full picture.

3.10.5 PROPOSITION.

(1) *The inequalities are equivalent.*

(2) *The equalities are equivalent if T_H and T_L preserve stability.*

(3) *The first (second) equality implies T_H (T_L) preserves stability.*

PROOF: We shall only prove one half in each case, the other halves being exactly analogous.

(1)

$$\begin{aligned} down\ T_H h &\geq down\ T_H down\ up\ h \\ &\geq down\ up\ T_L down\ h \\ &\geq T_L down\ h \end{aligned}$$

(2) Let h be stable. Then

$$\begin{aligned} down\ T_H h &= down\ T_H down\ up\ h \\ &= down\ up\ T_L down\ h \\ &= T_L down\ h \end{aligned}$$

because *down* h is stable and T_L preserves stability.

(3) Let h be stable. Then

$$\begin{aligned} up\ down\ T_H h &= up\ down\ T_H up\ down\ h \\ &= up\ down\ up\ T_L down\ h \\ &= up\ T_L down\ h \\ &= T_H up\ down\ h \\ &= T_H h \end{aligned}$$

∎

Hence, to prove that T_H and T_L match, it suffices to prove one inequality plus corresponding equality for stability, and that the other T is stable.

3.10.6 LEMMA. *If the doublet is coincident and T_H, T_L match they are parallel.*

PROOF: Let T_L be from x, y to x', y'. Let $l : x \to y \in L$. Then $up\ l : x \to y$ and $up\ T_L l \parallel T_H up\ l$, which implies that T_H is from x, y to, say, x'', y''. Let $h : x \to y \in H$, so that

$$\begin{aligned} T_H h &\parallel T_H up\ l \\ &\parallel up\ T_L l \\ &\parallel T_L l \end{aligned}$$

Thus $x'' = x'$ and $y'' = y'$. ∎

Now let $K_H < H$ and $K_L < L$, and let $F_H : H \to H$, $F_L : L \to L$ be a K_H and K_L functor respectively. F_H, F_L *match* when appropriate corresponding bundles are matching translators, ie: for any objects $x, y \in H$, $F_H[x, y]$ and $F_L[down\ x, down\ y]$ are matching translators. The same comments apply to preserving stability.

Furthermore, if the doublet is coincident, F_H and F_L agree on objects.

3.10.7 LEMMA (MATCHING ITERATION). *Given matching recursors or functors, X_H, X_L on H, L respectively, and matching arrows h, l (to which X_H, X_L are respectively applicable), then for any $n \geq 0$*

$$down(X_H^n h) = X_L^n l$$

$$up(X_L^n l) = X_H^n h$$

Moreover, each iterate is stable.

PROOF: By induction on n. Assume true for n: then both $X_H^n h$, $X_L^n l$ are stable. Thus

$$\begin{aligned} down(X_H^{n+1} h) &= down\ X_H(X_H^n h) \\ &= X_L\ down(X_H^n h) \\ &= X_L(X_L^n l) \\ &= X_L^{n+1} l \end{aligned}$$

and

$$\begin{aligned} up(X_L^{n+1} l) &= down\ X_L(X_L^n l) \\ &= X_H\ down(X_L^n l) \\ &= X_H(X_H^n h) \\ &= X_H^{n+1} h \end{aligned}$$

Clearly the result holds for $n = 0$ — this is just the matching of h, l. ∎

3.10.8 COROLLARY. *Let R_H, R_L be matching recursors, with matching seeds a_H, a_L. Then $\mathrm{fix}_{a_H} R_H$ and $\mathrm{fix}_{a_L} R_L$ match. Hence both LFPs are stable.*

PROOF: Apply the Lemma, and take sups. ∎

We can also define the idea of *matching joins(sums)*, being matching functors for which the injections also match.

Matching Limits.

First, suppose that $[\![\cdot]\!]_H$, $[\![\cdot]\!]_L$ are matching adjunctive diagrams (over the same net N, in H, L respectively). Then, since *up*, *down* are thread functors:

- If $[\![\cdot]\!]_H \triangleright i_H$ is a (co)limit, so is $down[\![\cdot]\!]_H \triangleright down\ i_H$
- If $[\![\cdot]\!]_L \triangleright i_L$ is a (co)limit, so is $up[\![\cdot]\!]_L \triangleright up\ i_L$

Thus $up\ down\ i_H$, $down\ up\ i_L$ are stable limits . We shall call them *matching limits*.

Second, let $F_H : H \to H$, $F_L : L \to L$ be matching functors (a K_H- and K_L-functor respectively), with matching seeds $k_H : x \to F_H x$, $k_L : y \to F_L y$. Then the Matching Iteration Lemma guarantees that $[\![\cdot]\!]_{k_H, F_H}$ matches $[\![\cdot]\!]_{k_L, F_L}$, whereupon we can proceed as above to establish matching limits.

Matching Solutions.

Any lower functor will preserve relative injections (and a fortiori relative isos). Hence $up, down$ will do so; furthermore, if $h : p_H \cong q_H \in H$, we have

$$\begin{aligned} up\ down\ p_H &\leq up\ down\ h\,;up\ down\ q_H\ ;up\ down\ h^- \\ &\leq h\,;up\ down\ q_H\ ;h^- \end{aligned}$$

and likewise

$$up\ down\ q_H \leq h^-\ ;up\ down\ p_H\ ;h$$

So $h : up\ down\ p_H \cong up\ down\ q_H$. In particular, if one of p_H, q_H is stable, so is the other.

The same argument is valid in L *provided* that the *iso* is stable.

A stable solution in one half of the doublet will create a solution in the other half. And if the first is generated by a stable seed, the second will be generated by the corresponding seed. The precise picture is as follows.

Let F_H, F_L be matching (monadic) K_H, K_L-functors. Let $f_H : x_H \to F_H x_H, f_L : x_L \to F_L x_L$ be media in H, L respectively (if one is a projection, both are because $up, down$ are thread functors).

3.10.9 THEOREM. *$\binom{f_H}{f_H}$ and $\binom{f_L}{f_L}$ are matching recursors on Px_H and Px_L.*

PROOF: First, note that for any medium $f : x \to Fx$ and $p \leq x$, f, Fp fit, as do Fp, f^-, because $f^R = 1$ and $Fp \leq 1$. So any thread functor T will have $T\left(\binom{f}{f}_F p\right) = \binom{Tf}{Tf}_{TF} p$. Thus

$$\begin{aligned} up\left(\begin{pmatrix} f_L \\ f_L \end{pmatrix}_{F_L} p_L\right) &= \begin{pmatrix} f_H \\ f_H \end{pmatrix}_{upF_L} p_L \\ &\leq \begin{pmatrix} f_H \\ f_H \end{pmatrix}_{F_H up} p_L \end{aligned}$$

because $up\ F_L p_L \leq F_H up\ p_L$ with $=$ if p_L is stable

$$= \begin{pmatrix} f_H \\ f_H \end{pmatrix}_{F_H} p_H$$

and similarly

$$\begin{aligned} down\left(\begin{pmatrix} f_H \\ f_H \end{pmatrix}_{F_H} p_H\right) &= \begin{pmatrix} f_L \\ f_L \end{pmatrix}_{downF_H} p_H \\ &\geq \begin{pmatrix} f_L \\ f_L \end{pmatrix}_{F_L} p_L \quad \text{with} = \text{if } p_H \text{ is stable} \end{aligned}$$

∎

3.10.10 THEOREM. *Let $p_H \leq x_H, p_L \leq x_L$ be matching. Then*

(1) *If a_H, a_L are matching seeds for $\binom{f_H}{f_H}$ and $\binom{f_L}{f_L}$, then a_H generates p_H iff a_L generates p_L.*

(2) p_H *is a a solution iff* p_L *is.*

PROOF:

(1) We know that $f_H /_{a_H} f_H$ matches $f_L /_{a_L} f_L$, so $a_H \leq p_H \leq f_H /_{a_H} f_H$ iff $a_L \leq p_L \leq f_L /_{a_L} f_L$

(2) If p_H is a solution of F_H, then

$$p_H \leq \binom{f_H}{f_H} p_H \Rightarrow p_L \leq down \binom{f_H}{f_H} p_H = \binom{f_L}{f_L} p_L$$

and

$$F_H p_H \leq f_H^- ; p_H ; f_H \Rightarrow F_L p_L \leq f_L^- ; p_L ; f_L$$

The converse is analogous. ∎

3.11 CONSTRUCTORS

In this section we look at a way of extending an object function over a Set-like bicontext to a functor. The function must systematically *construct*, in a certain sense, the underlying set of an image object from those of its arguments objects.

Let C be a Set-like bicontext, with $K \trianglelefteq^{=} C$ spanning and pre-compact. Let $X = \text{obj}C$. A *K-constructor* is a function $F : X^I \to X$ (arity I), together with a set $\text{Op} = \bigcup_{\tau \in I^n} \text{Op}_\tau$ of *operators* of various *sorts* τ (if $\tau \in I^n$, $n = |\tau|$), a set of *terms*, $\text{Term} = \bigcup_{\substack{\tau \in I^n \\ x \in X^I}} \text{Term}_{\tau,x}$ where $\text{Term}_{\tau,x} = \text{Op}_\tau \times x^\tau$ and $x^\tau = x_{\tau_1} \times \cdots \times x_{\tau_n}$, and an onto partial-function, $[\![\cdot]\!] = \bigcup_{\substack{\tau \in I^n \\ x \in X^I}} [\![\cdot]\!]_{\tau,x}$ with $[\![\cdot]\!]_{\tau,x} : term_{\tau,x} \rightharpoonup Fx$.

The set of names is $\text{Name} = \bigcup_{\substack{\tau \in I^n \\ x \in X^I}} \text{Name}_{\tau,x}$, where $\text{Name}_{\tau,x} = ![\![\cdot]\!]_{\tau,x}$. For term $oa \in \text{Term}_{\tau,x}$, define $|oa| = \{a_j \mid j = 1, ..., |\tau|\}$ and $|oa|_i = \{a_j \mid \tau_j = i\}$.

The import of this is that the elements of Fx can be represented in a systematic way in terms of the component x_i.

Now let $x, y \in X^I$ and $A \subseteq C^I$. Consider tuples $t = \langle o_1 a_1, ..., o_m a_m \rangle \in \text{Term}_{\tau_1,x} \times \cdots \times \text{Term}_{\tau_m,x}$ and $t' = \langle o_1 b_1, ..., o_m b_m \rangle \in \text{Term}_{\tau_1,y} \times \cdots \times \text{Term}_{\tau_m,y}$ (each $a_j \in$

x^{τ_i}, $b_j \in y^{\tau_i}$) — write $t \equiv_m t'(x,y)$. These induce the finite relation

$$\{\langle a_{jk}, b_{jk}\rangle \mid j = 1, ..., m \ \& \ k = 1, ..., |\tau_j|\} \subseteq \bigcup_{i,j}(|t_j|_i \times |t'_j|_i)$$

because each $a_{jk} \in x_{\tau_{jk}}$ and $b_{jk} \in y_{\tau_{jk}}$

If the I-tuple of restrictions of this relation to each $\bigcup_{i,j}(|t_j|_i \times |t'_j|_i)$ (the i^{th} *induced relation*) is in $\langle A|$, we shall say that t, t' *satisfy* A (in case $A = A_0^I$ for $A_0 \subseteq C$, we shall say simply "satisfy A_0"). Notice that the tuples must be *congruent* in the sense that the corresponding terms must have the same operator. We shall call $\bigcup_j |\tau_j|_i$ the i^{th} *component* of t.

We now impose two conditions on a constructor. The first requires that the "K-structure" of Fx is determined by those of the x_i in the sense that, given tuples $\langle [o_1 a_1], ..., [o_m a_m]\rangle$ from Fx and $\langle [o_1 b_1], ..., [o_m b_m]\rangle$ from Fy, if the a's "look like" the b's (with respect to K), then the two tuples will look alike. The second requires that namehood is similarly determined. Thus:

(C_F) If $t \equiv_m t'(x,y)$ satisfy K, then $\{\langle [t_j], [t'_j]\rangle \mid j = 1, ..., m\} : Fx \to Fy$ is in K

(C_N) If $t \equiv_1 t'(x,y)$ satisfy K and $t \in \mathrm{Name}_{r,x}$, then $t' \in \mathrm{Name}_{r,y}$

We now extend F to C^I as follows. For $r : x \to y \in C$ define

$$Fr = \{\langle [t], [t']\rangle \mid t \equiv_1 t'(x,y) \text{ satisfy } \{r\}\}$$

Thus $[oa]\ Fr\ [o'b]$ when $o = o' \in \mathrm{Op}_\tau$ and every $a_j\ r_{\tau_i}\ b_j$ ($i \in I, j = 1, ..., |\tau|$).

3.11.1 PROPOSITION.

(1) *$Fr : Fx \to Fy$ in C*

(2) $F1 = 1$

(3) *$r \leq s$ implies $Fr \leq Fs$*

(4) $F(r^-) = (Fr)^-$

So F becomes a monotonic symmetric pre-functor. But if $D \nearrow r$, then t, t' satisfy r iff they satisfy some $d \in D$, because the induced relation is finite, so that:

(a) each i^{th} induced relation is finite

(b) all but finitely many i^{th} induced relations are empty

Hence F is also continuous, and we have

3.11.2 THEOREM. *F is a functor on C^I.*

Next we show that

3.11.3 THEOREM. *F is within K.*

PROOF: Let $u : x \to y \in K^I$ and let $[t_j]\ Fu\ [t'_j]$ $(j = 1, ..., m)$. Since the relation induced by $\langle t_j\rangle_j, \langle t'_j\rangle_j$ is the i-wise union (over $j = 1, ..., m$) of those induced by the t_j, t'_j, and each of the latter is $\leq u$, the former is also $\leq u$. Hence $\langle t_j\rangle_j, \langle t'_j\rangle_j$ satisfy K. Thus by C_F

$$\{\langle [t_j], [t'_j]\rangle \mid j = 1, ..., m\} \in K$$

So every finite $r \subseteq Fu$ is in K, whence $Fu \in K$. ∎

Thus F is a functor within K. The next theorem shows it is also K-lower

3.11.4 THEOREM. *F is K-lower.*

PROOF: Let $u : x \to y \in K^I$, $r : y \to z \in C^I$. We want $F(u\,;r) \leq Fu\,;Fr$. Let $[oa]\ F(u\,;r)\ [oc]$. For each $j = 1, ..., |\tau|\ o \in \mathrm{Op}_\tau)$, let $b_j \in y_{\tau_j}$ with $a_j\ u_{\tau_j}\ b_j\ r_{\tau_j}\ c_j$. This is possible because each $a_j\ u_{\tau_j}\,;r_{\tau_j}\ b_j$. But then oa, ob satisfy K, so ob is a name. Clearly

$$[oa]\ Fu\ [ob]\ Fr\ [oc]$$

and, of course, $ob \in \mathrm{Term}_{\tau,y}$, whence $[ob] \in Fy$. ∎

We turn to conditions for F to be exact.

3.11.5 LEMMA (EXACTNESS 1). *Let* $r : x \to y \in C^I$, $u : y \to z \in K^I$, *with every* $r_i! \subseteq !u_i$ *and* Fu *singular. Then* $Fr\,; Fu = F(r\,; u)$.

PROOF: We need only prove $Fr\,; Fu \leq F(r\,; u)$. Let $[oa]\ Fr\ [ob] = [o'b']\ Fu\ [o'c']$, $(o \in \mathrm{Op}_r, o' \in \mathrm{Op}_{r'})$. Then for $j = 1, ..., |r|$, $b_j \in r_{r_j}!$, so $b_j \in !u_{r_j}$, so let c_j be such that $b_j\ u_{r_j}\ c_j$. Then ob, oc satisfy K, whence oc is a name. Thus $[ob]\ Fu\ [oc]$. By the singularity, $[oc] = [o'c']$. But clearly $[oa]\ F(r\,; u)\ [oc]$, so that $[oa]\ F(r\,; u)\ [o'c']$. ∎

We now impose a further condition on the behaviour of names.

(C_D) F is *directed* when, given names $n_1 \in \mathrm{Name}_{r_1,x}, n_2 \in \mathrm{Name}_{r_2,x}$ with $[n_1] = [n_2] = b \in Fx$, there is a name $n \in \mathrm{Name}_{r,x}$ with $[n] = b$ and $|n| \subseteq |n_1| \cap |n_2|$.

3.11.6 LEMMA (EXACTNESS 2). *Let* $u : x \to y$, $v : y \to z \in K^I$, *with* Fu *co-singular and* Fv *singular. Then* $Fu\,; Fv = F(u\,; v)$.

PROOF: Again, we need only prove $Fu\,; Fv \leq F(u\,; v)$. Let $[o_1a_1]\ Fu\ [o_1b_1] = [o_2b_2]\ Fv\ [o_2c_2]$, $(o_1 \in \mathrm{Op}_{r_1}, o_2 \in \mathrm{Op}_{r_2})$. By directedness, choose name ob ($o \in \mathrm{Op}_r$) for the middle element, with $|ob| \subseteq |o_1b_1| \cap |o_2b_2|$. It is now possible to choose, for each $b_j \in |ob|$ $(j = 1, ..., |r|)$, $a_j \in |o_1a_1|$ and $c_j \in |o_2c_2|$ such that $a_j\ u_{r_j}\ b_j\ v_{r_j}\ c_j$. As in the previous proof, oa and oc are then names, $[oa]\ Fu\ [ob]\ Fv\ [oc]$ and $[oa]\ F(u\,; v)\ [oc]$. By the singularity conditions $[o_1a_1] = [oa]\ F(u\,; v)\ [oc] = [o_2c_2]$ ∎

By using second-level derived operators, it is possible to show that the composite of constructors is also one, but the details are rather tedious.

K-Alignment.

From the exactness Lemmata we now obtain

3.11.7 THEOREM (EXACTNESS). *If $K \subseteq \mathrm{Int}(C)$, directed F is exact on K^I.*

Assume now both the conditions of the Theorem, viz: that $K \subseteq \mathrm{Int}(C)$ and that F is directed. We shall show that the requirements for aligning F to K are met in this situation.

The only condition on F is that it be exact on K, which has just been proven. K itself must be pre-compact and spanning, which it is by hypothesis. The remaining requirement is an alternative: either K must be projective, or the target bicontext must be linear almost-complete, which it is.

Hence we can align F to K, and then, because $K \subseteq \mathrm{RegInt} = \mathrm{Int}$, we obtain

3.11.8 THEOREM. *F_K is a K-functor.*

Chapter 4

Induction

In this chapter we take a general view of the idea of inductive proof by isolating the concept of an *induction principle*. Not only do we want a foundation for the particular kinds of structural induction mentioned in Chapter 1, but also a general explanation of how inductions may be passed by functions from one set to another, possibly being combined in some way en route. One example is induction on the depth of a tree, say, where an induction principle on ω *creates* one on the relevant space of trees. Another might be "nested" induction on one variable "within" another, say proving $(\forall x, y)p(x, y)$ by "induction on y within x", wherein we tacitly use an induction principle on ω^2 created by combining one on ω with itself in a certain way.

Let us consider standard Mathematical Induction (MI) on ω as motivation for our general treatment. If $X \subseteq \omega$, MI says

$$0 \in X \;\&\; \forall n(n \in X \Rightarrow n+1 \in X) \Rightarrow X = \omega$$

Rephrase this as

$$\{0\} \cup \{n+1 \mid n \in X\} \subseteq X \quad \Rightarrow \quad X = \omega$$

and we can see that the crux of the induction principle is the function

$$MI : \mathrm{P}\omega \rightarrow \mathrm{P}\omega : X \mapsto \{0\} \cup \{n+1 \mid n \in X\}$$

which has the property that $\{X \subseteq \omega \mid FX \subseteq X\} = \{\omega\}$. Below we define a principle to be *any* function over subsets and an *induction* principle to be one with this particular property.

Another example on ω is Course of Values Induction (CoV), whose principle is

$$CoV : X \to \{n \mid X \ni \forall i < n\}$$

Notice that every $CoV(X) \subseteq MI(X)$; this captures the idea that CoV is better (stronger) than MI because it furnishes a stronger induction *hypothesis*. We shall also make a definition along these lines, though we base it rather upon the fact that $CoV(X) \cup X \subseteq MI(X) \cup X$, because clearly elements *already* in X do not matter.

One further generalisation is desirable. We wish to discuss structural induction in an object (x) of some bicontext, so Pω must give way to any such Px. Now Px has a top (x itself), and much of the initial theory can be couched solely in terms of an order with top — later it will become necessary to assume completeness of the order, but in the presence of top this is guaranteed by almost-completeness.

Having set the scene, we now turn to the technicalities.

Convention.

Throughout this chapter, $X, Y, A, B...$ (typical elements $x, y, a, b...$) will be types with a top, denoted by 1 (possibly annotated), and $C, ...$ will be bicontexts

4.1 DEFINITIONS

4.1.1 DEFINITION. *A* principle *on X is any function $Q : X \to X$.* Step$Q$ *is the subset $\{x \in X \mid Qx \leq x\}$. Q is an* induction *when* Step$Q = \{1\}$.

Notice that 1 is the only element which an induction can map into 0. As mentioned above, we shall be especially interested in principles on various Px, where x is an object in some bicontext. In such a case we shall refer to a principle on x, or more generally, if $p \leq q \leq x$, a principle on $[p, q]$, being one on $\{p' \in \mathrm{P}x \mid p \leq p' \leq q\}$.

4.1.2 DEFINITION. *For principles Q, Q' on X, define*

(1) $Q \wedge Q' : x \mapsto Qx \wedge Q'x$

(2) $Q \vee Q' : x \mapsto Qx \vee Q'x$

(3) $Q^+ = Q \vee X$

(we are using here the convention that X stands for its identity).

4.1.3 DEFINITION. *The* strength *preorder* $\succeq$ *(*no weaker than*) on principles is defined by*

$$Q \succeq Q' \Leftrightarrow Q^+ \leq Q'^+$$

*The corresponding equivalence (*equipotence*) is* $\simeq$.

We list some immediate properties.

- Step$(Q \vee Q') = StepQ \cap StepQ'$
- Step$X = X$
- Step$Q^+ =$ Step$Q$ is the set of fixed-points of Q^+
- If Q is monotonic, so is Q^+
- $Q^{++} = Q^+$
- If $Q \succeq Q'$, then Step$Q \supseteq$ StepQ', in which case Q' is an induction if Q is.
- The Q^+s are canonical representatives of the $\simeq$-classes.
- If X is an I-tuple of complete types with Q_i a principle on each X_i, then Step$\prod Q = \prod_{i \in I}$ StepQ_i and $\prod Q$ is an induction if each Q_i is.
- Step$Q$ is closed under infs.

When Q is an induction, the last property enables a collection of induction proofs to be combined into a single proof that the intersection of all the separate properties is universal.

4.2 P-PAIRS

In this and the next section we consider various relationships between principles on different complete types, embodied in functions between the orders. A function $f : X \to Y$ is *isolating* when $fx = 1 \Rightarrow x = 1$. Notice that the composite of isolating functions is also one.

4.2.1 DEFINITION. *A* P-pair *is a pair of functions* $s : X \leftrightarrows Y : s^*$ *such that*

$$s^* y \leq x \Rightarrow y \leq sx$$

It is isolating *when* s *is isolating.* s *is a* left conjugate *for* s^*, *which is a* right conjugate *for* s. *We may write the P-pair as* $s : X \Longrightarrow Y$.

The composite $\langle s\,;t,\ t^*\,;s^*\rangle$ of appropriate P-pairs is also one, and any $\langle X, X\rangle$ is obviously a P-pair, so the P-pairs form a category, indeed, a context taking the functions under pointwise ordering. It is, in fact, a sub of the pair-bicontext of the context comprising all (small) types and *any* functions between them (of which Type2 is also a bisub); it contains the isolating P-pairs as an inner sub.

The P-pair sub is not symmetric because in general $\langle s^*, s\rangle$ is not a P-pair — when it is we have a *bi*-P-pair. We can similarly talk of P-isos, etc.

4.2.2 PROPOSITION. *For the P-pair* $s : X \Longrightarrow Y$, $s1 = 1$ & $y \leq ss^*y$.

PROOF:

$$\begin{aligned} s^*1 \leq 1 &\Rightarrow 1 \leq s1 \\ &\Rightarrow s1 = 1 \end{aligned}$$

Also

$$s^*y \leq s^*y \Rightarrow y \leq ss^*y$$

∎

Examples.

(1) Let X be an I-tuple of complete types. For $i \in I$, $x_0 \in \prod X$, consider the component projection $(\cdot)_i : X \to X_i$ (non-isolating) and the superimposition function $x_0[i] : X_i \to X$ (isolating).

Then for any $x \in \prod X$, $x' \in X_i$, we have $x_i \leq x' \Rightarrow x \leq 1[i]x'$ so $1[i]$ is a left conjugate for $(\cdot)_i$. We shall take this P-pair as canonical from X_i to X.

Also, for $x' \in X_i$ again, $x_0[i]x' \leq x \Rightarrow x' \leq x_i$, so every $x_0[i]$ is a right conjugate for $(\cdot)_i$. Thus the P-pair above is symmetric.

(2) Any polyadic functor (of arity I) determines a function : $\prod_{i \in I} \mathrm{P}x_i \to \mathrm{P}(Fx)$ for any I-tuple x of objects. We can therefore talk about a left- or right-conjugate for F *at* x, referring to this function.

(3) Any function : $\{1\} \to X$ is a non-isolating right conjugate for the only function : $X \to \{1\}$, also non-isolating.

Now, given P-pair $s : X \Longrightarrow Y$ and principle Q on Y, define principle Q^s on X to be $s\,;Q\,;s^*$ (ie: $Q^s x = s^*(Q(sx))$). Then

4.2.3 LEMMA. $x \in \mathrm{Step}Q^s \Rightarrow sx \in \mathrm{Step}Q$.

PROOF: Let $Q^s x \leq x$. Then

$$\begin{aligned} s^*Q(sx) \leq x &\Rightarrow Q(sx) \leq sx \\ &\Rightarrow sx \in \mathrm{Step}Q \end{aligned}$$

∎

4.2.4 COROLLARY. *If Q is an induction and s is isolating, Q^s is an induction.*

We may say that the induction Q *creates* the induction Q^s (via s). It is immediate that for composable P-pairs s, t $Q^{s;t} = (Q^t)^s$, and that $Q^X = Q$.

The components of a P-pair may be monotonic; in this regard we have

4.2.5 PROPOSITION. *If the pair of functions $s : X \leftrightarrows Y : s^*$ has s monotonic, it is P-pair iff $s^* ; s \geq Y$.*

PROOF: The forward implication is just (4.2.2). Conversely

$$s^* y \leq x \;\Rightarrow\; s s^* y \leq s x$$

$$\Rightarrow\; y \leq s x$$

∎

Now let X, Y be complete, and suppose now that we have P-pairs $s_i : X \Longrightarrow Y_i$, $t_i : X_i \Longrightarrow Y$ for each $i \in I$. Define $s : X \leftrightarrows \prod_i Y_i : s^*$, $t : \prod_i X_i \leftrightarrows Y : t^*$ by

$$sx = \langle s_i x \rangle_i \qquad tx = \bigwedge_i t_i x_i$$

$$s^* y = \bigvee_i s_i^* y_i \qquad t^* y = \langle t_i^* y \rangle_i$$

Then

$$s^* y \leq x \;\Rightarrow\; s_i^* y_i \leq x \quad \forall i \in I$$

$$\Rightarrow\; y_i \leq s_i x \quad \forall i \in I$$

$$\Rightarrow\; y \leq sx$$

and

$$t^* y \leq x \;\Rightarrow\; s_i^* y \leq x_i \quad \forall i \in I$$

$$\Rightarrow\; y \leq t_i x_i \quad \forall i \in I$$

$$\Rightarrow\; y \leq tx$$

Thus $\langle s, s^* \rangle, \langle t, t^* \rangle$ are P-pairs. Moreover, suppose that some s_i and every t_i is isolating. Then

$$sx = 1 \;\Rightarrow\; s_i x = 1 \quad \forall i \in I$$

$$\Rightarrow\; x = 1$$

so s is isolating, and

$$tx = 1 \;\Rightarrow\; t_i x_i = 1 \quad \forall i \in I$$

$$\Rightarrow x = 1$$

so t is isolating.

If $Y = \prod_i X_i$ and t is the canonical P-pair $\langle 1[i], (\cdot)_i \rangle$ above, then $t = t^* = Y$. Reverting to the general case, we have for each $i \in I$ since $t_i 1 = 1$

$$\begin{aligned} t(1[i]x_i) &= \bigwedge_{j \in I} (\text{if } j = i \text{ then } t_i x_i \text{ else } 1) \\ &= t_i x_i \end{aligned}$$

and

$$(t^* y)_i = t_i^* y$$

Thus the t_i factor through the $\langle 1[i], (\cdot)_i \rangle$, but not uniquely in general.

Projective P-pairs.

Let $s^* : Y \to X$ be a projective function with $s^* 1 = 1$. Define $s : X \to Y$ by

$$sx = \text{ top of } \{y \in Y \mid s^* y \leq x\}$$

Then

$$s^* y \leq x \Rightarrow y \leq sx$$

and

$$\begin{aligned} sx = 1 &\Rightarrow s^* 1 \leq x \\ &\Rightarrow x = 1 \end{aligned}$$

so $\langle s, s^* \rangle$ is an isolating P-pair.

A prime example of this is to create an induction on some integer measure over a set, such as the length of a sequence or the depth of a tree. In such a case the relevant bicontext will be Set or Part(Set), and X, Y will be some $\mathrm{P}x$ and $\mathrm{P}\omega$ respectively. For $q \in \mathrm{P}\omega$, $s^* q$ will be the inverse image of $\{x \mid \text{measure } x \in q\}$.

Since s^* is linear it is projective if $\mathrm{P}x$ is complete, and obviously satisfies $s^*y = x$. Any induction principle on ω then creates one on x.

Other examples of this kind, associated with functors, will appear later.

4.3 Nested Inductions

In this section we define a derived induction which represents the nesting of one induction "within" another.

4.3.1 DEFINITION. *The set* $\{s_x : V \Longrightarrow Y \mid x \in X\}$ *of P-pairs, indexed by the type* X, *is a* P-family *for the function* $s : V \to X$ *when for any* $x \in X,\ v \in V$ *we have*

$$s_x v = 1 \Rightarrow x \leq sv$$

Notice that s_1 is isolating if s is.

Given principles Q^X, Q^Y on X, Y respectively, the principle Q^s on V (here the s stands for the whole family) is defined by

$$Q^s v = (Q^Y)^{s_x} v, \quad \text{where } x = Q^X(sv)$$

A P-pair can be composed with the P-family in the sense that, if $t : W \Longrightarrow V$ is a P-pair, $\{t\,;s_x \mid x \in X\}$ is a P-family for $t\,;s$.

4.3.2 THEOREM. *If* Q^X, Q^Y *are inductions and* s *is isolating,* Q^s *is an induction.*

PROOF:

$$\begin{aligned}
v \in \mathrm{Step}Q^s &\Rightarrow v \in \mathrm{Step}Q^{Y,s_x} \\
&\Rightarrow s_x v \in \mathrm{Step}Q^Y \\
&\Rightarrow s_x v = 1 \\
&\Rightarrow x \leq sv \\
&\Rightarrow sv \in \mathrm{Step}Q^X \\
&\Rightarrow sv = 1 \\
&\Rightarrow v = 1
\end{aligned}$$

Again, we may say that Q^X, Q^Y create Q^s via s.

As we remarked above, Q^s represents induction on "Q^Y within Q^X". We shall show this more clearly in a later example (it is possible to extend the family concept to n-ary families, but the formulation is rather messy).

Notice that since s_1 is isolating, it can also be used alone to create the induction Q^{Y,s_1}. The two inductions are generally incomparable for strength, but we would expect Q^s to be better in some sense. We shall return to this point again later.

4.3.3 EXAMPLES.

We now present two important examples of nested inductions. Prima facie, they may seem contrived; their significance will appear below.

Example 1.

Define

$$s : A \times B \to A : \langle a, b \rangle \mapsto \text{if } b = 1 \text{ then } a \text{ else } 0$$

and for $a \in A$

$$s_a : A \times B \to B : \langle a', b' \rangle \mapsto \text{if } a' \geq a \text{ then } b' \text{ else } 0$$

$$s_a^* : B \to A \times B : b \mapsto \langle a, b \rangle$$

Then s is isolating and each $\langle s_a, s_a^* \rangle$ is a P-pair because

$$s_a^* b \leq \langle a', b' \rangle \;\Rightarrow\; a \leq a' \;\&\; b \leq b'$$

$$\Rightarrow\; b \leq s_a(a', b') = b'$$

Also

$$s_a(a', b') = 1 \;\Rightarrow\; a \leq a' \;\&\; b' = 1$$

$$\Rightarrow a \leq a' = s(a', b')$$

Thus $\{\langle s_a, s_a^* \rangle \mid a \in A\}$ is a P-family for s, so inductions Q^A, Q^B on A, B respectively create the induction Q^s on $A \times B$ (for the remainder of the example called simply Q). Then

$$\begin{aligned} Q(a,b) &= s_{a_0}^* Q^B s_{a_0}(a,b) \quad \text{where } a_0 = Q^A s(a,b) \\ &= \langle a_0, Q^B s_{a_0}(a,b) \rangle \end{aligned}$$

In particular, we have

$$\begin{aligned} Q(1,1) &= \langle Q^A 1, Q^B 1 \rangle \\ Q(a < 1, 1) &= \langle Q^A a, Q^B 0 \rangle \\ Q(a \geq Q^A 0, b < 1) &= \langle Q^A 0, Q^B b \rangle \\ Q(a \not\geq Q^A 0, b < 1) &= \langle Q^A 0, Q^B 0 \rangle \end{aligned}$$

Example 2.

Let $A \wr B$ be the set of *inverse* monotonic functions f from A to B, that is to say, monotonic functions from A to the dual order of B, with the property that every $f^{-}[b\rangle$ is $\langle a]$ for some $a \in A$ (whence $f0 = 1$).

This implies that $\forall b \in B$, $\exists$ greatest $a \in A$ with $fa \geq b$ — call it b^f. Then $fa \geq b \Leftrightarrow b^f \geq a$. Now $b \leq b' \Rightarrow b^f \geq b'^f$, so $(\cdot)^f \in B \wr A$. Therefore

- $(\cdot)^{(\cdot)^f} = f$
- $(fa)^f \geq a$
- $f(b^f) \geq b$
- $\langle f, (\cdot)^f \rangle$ is a Galois connection between A and B.

Conversely, if $f : A \leftrightarrows B : f'$ is a Galois connection, each is inverse monotonic and $f' = (\cdot)^f$. Furthermore, if $f \leq g$ with respect to the pointwise ordering, $(\cdot)^f \leq (\cdot)^g$. Hence $A \wr B$ can be identified with the set of (left-halves of) Galois connections between A and B.

Obviously $\lambda a.1_B$ is top in $A \wr B$. If $H \subseteq A \wr B$ and $\forall a \in A$, $\bigwedge_{h\in H} ha$ exists, then $f \stackrel{\text{def}}{=} \lambda a. \bigwedge_{h\in H} ha = \bigwedge H$, because $fa \geq b \Leftrightarrow ha \geq b$, $\forall h \in H$, ie:

$$\begin{aligned} f^{-}[b\rangle &= \bigcap_{h\in H} h^{-}[b\rangle \\ &= \bigcap_{h\in H} \langle a_h] \quad \text{(say)} \\ &= \left\langle \bigwedge_{h\in H} a_h \right] \end{aligned}$$

In particular, $A \wr B$ is complete if B is.

Given $a_1 \leq \cdots \leq a_n \in A$ and $b_1 \geq \cdots \geq b_n \in B$, define $[a_1 \to b_1, \ldots, a_n \to b_n]$ to be the function

$$\lambda x.\ \text{if } x = 0 \text{ then } 1 \text{ elseif } x \leq a_1 \text{ then } b_1 \text{ else} \ldots \text{if } x \leq a_n \text{ then } b_n \text{ else } 0$$

Clearly this function is in $A \wr B$. For to calculate $a = b^{[\cdots]}$, place b as far to the right as possible in the sequence

$$1 = b_0 \geq b_1 \geq \cdots \geq b_n \geq b_{n+1} = 0,$$

say $b_i \geq b > b_{i+1}$. Then $a = a_i$ in the sequence

$$0 = a_0 \leq a_1 \leq \cdots \leq a_n.$$

Now consider the functions

$$1^{(\cdot)} : A \wr B \to A$$

and for any $a \in A$

$$(\cdot)a : A \wr B \leftrightarrows B : [a \to \cdot]$$

Then

$$1^f = 1 \Rightarrow f = 1 \text{ (ie : } \lambda a.1)$$

$$fa = 1 \Rightarrow a \leq 1^f$$

and

$$[a \to b] \leq f \;\Rightarrow\; b \leq fa$$

Hence the set $\{\langle(\cdot)a, [a \to \cdot]\rangle \mid a \in A\}$ is an isolating P-family for $1^{(\cdot)}$, so from inductions Q^A, Q^B on A, B respectively it creates the induction Q on $A \wr B$ with

$$\begin{aligned} Qf &= [a \to Q^B fa] \qquad \text{where } a = Q^A 1^f \\ &= [Q^A 1^f \to Q^B f(Q^A 1^f)] \end{aligned}$$

Henceforth in this chapter, all types will be complete and all bicontexts locally-complete.

4.4 Paths

In this section we associate with every induction principle on a complete type an ordinal-tuple within it, and vice versa. This association then enables a connection to be established between induction via a principle and standard transfinite induction. It also serves to justify the terminology "nested induction".

4.4.1 Definition. *A* path *in complete type X is a tuple $p \in X^{\mathrm{Ord}}$. Define*

$$\bar{p}_\xi = \bigvee_{\chi < \xi} p_\chi$$

Note that $\bar{p}_0 = 0$. Because X is a set, there are two ordinals, say $\chi < \xi$ for which $p_\xi = p_\chi$, whence $p_\xi \leq \bar{p}_\xi$. Let α be the smallest such ξ. We call this the *length* of p, and refer to p as a $\bar{p}_\alpha$-path.

4.4.2 Definition. *Paths p, q are* equipotent, *$p \equiv q$, when they have the same length (α) and*

$$p_\xi = q_\xi \quad \forall \xi < \alpha$$

4.4.3 Proposition. *If p has length α and $p_\xi = q_\xi$, $\forall \xi \leq \alpha$, then $p \equiv q$.*

Proof: Let q have length β. Clearly $\bar{p}_\xi = \bar{q}_\xi$, $\forall \xi \leq \alpha$, so $\beta \leq \alpha$. But if $\beta < \alpha$, we have $p_\beta \leq \bar{p}_\beta$, contradiction. ∎

Now let Q be a principle on X. We associate with it a path, also called Q — this will not normally cause confusion, because in use the path will nearly always have an ordinal subscript; if distinction is necessary the path will be referred to as $p(Q)$ — and defined by transfinite induction (TI) thus:

$$Q_\xi = Q(\bar{Q}_\xi)$$

The length of this path we shall call simply the length of Q. Note that, since $\bar{Q}_0 = 0,\ Q_0 = Q0$. The following proposition is immediate.

4.4.4 PROPOSITION. *If Q is an induction, $p(Q)$ is a 1-path.*

Conversely, given a path p on X of length α, we can define a principle $\Omega(p)$ (Ω for short) by

$$\Omega x = \begin{cases} \text{first } p_\xi \not\leq x \text{ with } \xi < \alpha, \text{ if such exists} \\ p_\alpha, \quad \text{else} \end{cases}$$

4.4.5 THEOREM. $p(\Omega(p)) \equiv p$

PROOF: By TI: assume $\Omega_\chi = p_\chi,\ \forall \chi < \xi$. Then

$$\begin{aligned} \Omega_\xi &= \Omega(\bar{\Omega}_\xi) \\ &= \Omega(\bar{p}_\xi) \\ &= \begin{cases} p_\xi, & \xi < \alpha \\ p_\alpha, & \xi = \alpha \end{cases} \end{aligned}$$

∎

4.4.6 PROPOSITION. *If p is a 1-path, $\Omega(p)$ is an induction.*

PROOF:

$$\begin{aligned} \Omega x \leq x &\Rightarrow p_\xi \leq x \quad \forall \xi < \alpha \\ &\Rightarrow \bar{p}_\alpha \leq x \\ &\Rightarrow x = 1 \end{aligned}$$

∎

In general there is not much connection between $\Omega(p(Q))$ and Q. But if we define Q to be *conservative* when $x \leq x'$ implies $Qx \leq x'$ or $x' \vee Qx = x' \vee qx'$, we do have

4.4.7 THEOREM. *If Q is a conservative induction, then $\Omega(p(Q)) \simeq Q$.*

PROOF: For $x < 1$, $\Omega(p(Q))x =$ first $p_\xi \not\leq x$. Then $\bar{p}_\xi \leq x$, so $x \vee Q\bar{p}_\xi = x \vee Qx$ since $p_\xi = Q\bar{p}_\xi \not\leq x$. Thus $\Omega^+ x = Q^+ x$. ∎

The significance of a conservative induction, when applied to some Set-like object in a bicontext, is that it adds to its argument set as little as is required to enlarge it, and this in a uniform way independently of what else the argument happens to contain. It thus captures the "first element not in the set" idea of CoV induction on ω, which is indeed conservative in contrast to MI, which generally adds unnecessary points and is not conservative.

We also have a connection when Q^+ is monotonic.

4.4.8 PROPOSITION. *If Q^+ is a monotonic induction, then $\Omega(p(Q)) \succeq Q$.*

PROOF: For $x < 1$, $\Omega x = p_\xi \not\leq x$ with $\bar{p}_\xi \leq x$, giving

$$\begin{aligned}\Omega^+ x &= x \vee p_\xi \\ &= x \vee \bar{p}_\xi \vee Q\bar{p}_\xi \\ &= x \vee Q^+\bar{p}_\xi \\ &\leq x \vee Q^+ x \\ &= Q^+ x\end{aligned}$$

∎

If Q is an induction on object x of some Set-like bicontext, we call it *singleton* to mean that for every $p \in \mathrm{P}x$, $Qp \setminus p$ is singleton. CoV on ω is singleton, although MI is not. Another example on ω is

$$\mathrm{CoV}_2 : p \mapsto \{\text{smallest even } \notin p \text{ if such exists, else smallest odd}\}$$

Now if Q is a singleton induction on any object x in Set, it generates a well-ordering of x via its path by

$$x_\xi = Q_\xi \setminus \bar{Q}_\xi$$

Since $\bar{Q}_\alpha = x$, where α is the length of Q, it is easy to see that this well-ordering of x is of order-type α. Thus, for example, CoV generates the natural well-ordering of type ω; and CoV_2 generates a well-ordering of order-type $\omega + \omega$.

We now calculate the paths for the nested induction examples of (4.3.3). In each case let Q^A, Q^B, Q have lengths α, β (> 0) and γ respectively.

Example 1.

We shall prove by TI that

(1) $$Q_\xi = \langle Q_0^A, Q_\xi^B \rangle \quad \xi < \beta$$

(2) $$Q_{\beta+\xi} = \langle Q_{1+\xi}^A, Q_0^B \rangle \quad \xi < \alpha$$

whence for $\xi \leq \alpha$

$$\begin{aligned} \bar{Q}_{\beta+\xi} &= \langle \bigvee_{\chi < 1+\xi} Q_\chi^A, \bigvee_{\chi<\beta} Q_\chi^B \rangle \\ &= \langle \bar{Q}_{1+\xi}^A, 1 \rangle \end{aligned}$$

giving $\gamma = \beta + \alpha$ (unless α is finite when $\gamma = \beta + (\alpha - 1)$).

Thus Q in effect performs Q^B followed by Q^A in sequence. This is reasonable, because if $A = \mathrm{P}x, B = \mathrm{P}y$ for objects $x, y \in C$, $A \times B$ represents pairs of independent properties, one on x and one on y. It is like a binary predicate $r(x, y)$ of the form $p(x)$ & $q(y)$ — a proof that r holds universally amounts to the catenation of corresponding proofs for p and q.

At this point we can compare Q^{s_1}:

$$Q^{s_1}(1, b) = \langle 1, Q^B b \rangle$$

$$Q^{s_1}(a < 1, b) = \langle 1, Q^B 0 \rangle$$

Thus Q^{s_1} obviously has length β. It makes no use of Q^A at all; the induction step must include an independent proof of $(\forall x)p(x)$. To this extent Q^{s_1} is worse than Q.

We now prove (1) and (2) by TI:

(1) $\xi < \beta$: for $\xi > 0$, $\bar{Q}_\xi = \langle Q_0^A, \bar{Q}_\xi^B \rangle$, so

$$Q_\xi = Q\ \bar{Q}_\xi$$

$$= s_a^*(Q^B(s_a(Q_0^A, \bar{Q}_\xi^B))) \quad \text{where} \quad \begin{cases} a = Q^A(s(Q_0^A, \bar{Q}_\xi^B)) \\ \quad = Q^A 0 \\ \quad = Q_0^A \end{cases}$$

so

$$Q_\xi = \langle Q_0^A, Q^B \bar{Q}_\xi^B \rangle$$

$$= \langle Q_0^A, Q_\xi^B \rangle$$

since $Q_\xi^B < 1$. And

$$Q_0 = Q(0,0)$$

$$= \langle Q_0^A, Q^B 0 \rangle$$

$$= \langle Q_0^A, Q_0^B \rangle$$

(2) $\xi < \alpha$: $\bar{Q}_{\beta+\xi} = \langle \bar{Q}_{1+\xi}^A, 1 \rangle$, so

$$Q_{\beta+\xi} = Q\ \bar{Q}_{\beta+\xi}$$

$$= s_a^*(Q^B(s_a(\bar{Q}_{1+\xi}^A, 1))) \quad \text{where} \quad \begin{cases} a = Q^A(s(\bar{Q}_{1+\xi}^A, 1)) \\ \quad = Q^A\ \bar{Q}_{1+\xi}^A \\ \quad = Q_{1+\xi}^A \end{cases}$$

so

$$Q_{\beta+\xi} = \langle Q_{1+\xi}^A, Q^B 0 \rangle$$

$$= \langle Q_{1+\xi}^A, Q_0^B \rangle$$

since $Q^B 0 < 1$ because $\beta > 0$.

Example 2.

Recall that

$$Qf = [Q^A 1^f \to Q^B(f(Q^A 1^f))]$$

We prove by TI that for $\xi < \alpha,\ \chi < \beta$,

$$Q_{\beta\xi+\chi} = [Q_\xi^A \to Q_\chi^B]$$

Assume true for all $\xi' < \xi$ with any χ', and for ξ with any $\chi' < \chi$. First we must calculate $\bar{Q}_{\beta\xi+\chi}$. Now if $a \in A, b \in B,\ A' \subseteq A, B' \subseteq B$

$$\bigvee_{b' \in B'} [a \to b'] = \left[a \to \bigvee B'\right]$$

and

$$\bigvee_{a \in A'} [a' \to b] = \left[\bigvee A' \to b\right]$$

So

$$\begin{aligned} \bar{Q}_{\beta\xi} &= \bigvee_{\xi' < \xi} [Q_{\xi'}^A \to 1] \\ &= [\bar{Q}_\xi^A \to 1] \end{aligned}$$

and

$$\begin{aligned} \bigvee_{\chi' < \chi} Q_{\beta\xi+\chi'} &= \bigvee_{\chi' < \chi} [Q_\xi^A \to Q_{\chi'}^B] \\ &= [Q_\xi^A \to \bar{Q}_\chi^B] \end{aligned}$$

Hence

$$\begin{aligned} \bar{Q}_{\beta\xi+\chi} &= [\bar{Q}_\xi^A \to 1] \vee [Q_\xi^A \to \bar{Q}_\chi^B] \\ &= [\bar{Q}_\xi^A \to 1,\ \bar{Q}_{\xi+1}^A \to \bar{Q}_\chi^B] \end{aligned}$$

and

$$1^{\bar{Q}_{\beta\xi+\chi}} = \bar{Q}_\xi^A \quad \text{because } \bar{Q}_\chi^B < 1$$

Thus

$$\begin{aligned} Q_{\beta\xi+\chi} &= Q\ \bar{Q}_{\beta\xi+\chi} \\ &= [Q_\xi^A \to Q^B(\bar{Q}_{\beta\xi+\chi}\ Q_\xi^A)] \end{aligned}$$

But $Q_\xi^A = Q^A \bar{Q}_\xi^A \not\leq \bar{Q}_\xi^A$ because $\bar{Q}_\xi^A < 1$. So

$$\bar{Q}_{\beta\xi+\chi}\ Q_\xi^A = \bar{Q}_\chi^B$$

whence

$$Q_{\beta\xi+\chi} = [Q^A_\xi \to Q^B_\chi]$$

The calculation of $\bar{Q}_{\beta\xi}$ is obviously valid for $\bar{Q}_{\beta\alpha}$, giving the value $[1 \to 1] = 1$. Thus $\gamma \leq \beta\alpha$. But for $\xi < \alpha$, $\chi < \beta$ we have

$$\bar{Q}_{\beta\xi+\chi} = 1 \Rightarrow \begin{cases} \text{either } \bar{Q}^A_\xi = 1 \Rightarrow \xi = \alpha \\ \\ \text{or } \bar{Q}^A_{\xi+1} = 1 \;\&\; \bar{Q}^B_\chi = 1 \Rightarrow \xi + 1 = \alpha \;\&\; \chi = \beta \end{cases}$$

contradiction.

Hence $\gamma = \beta\alpha$, and Q is, in a sense, a "free" lexicographic induction, on B within A.

Again, if we compare Q^{s_1}, which here is $f \mapsto [1 \to Q^B(f1)]$, we find that it too has length β. It represents a proof of, say, $(\forall x, y)p(x, y)$ by proving $\lambda y.(\forall x)p(x, y)$ universal using induction on y only. As in Example 1, it is worse in this sense than Q.

It is useful to know when a P-pair will create an induction of the same length as the original.

Suppose $s : X \Longrightarrow Y$ is an isolating P-pair, and let Q be an induction on Y. Assume that s has the following property:

$$\text{(S)} \qquad s\left(\bigvee_{\chi<\xi} s^*Q_\chi\right) = \bar{Q}_\xi \qquad \forall \xi \in \mathrm{Ord}$$

Then we have

4.4.9 THEOREM. *For any ξ, $s\bar{Q}^s_\xi = \bar{Q}_\xi$.*

PROOF: By TI. Assume true for all $\chi < \xi$. Then

$$\begin{aligned} s\bar{Q}^s_\xi &= s \bigvee_{\chi<\xi} Q^s(\bar{Q}^s_\chi) \\ &= s \bigvee_{\chi<\xi} s^*(Q(s\bar{Q}^s_\chi)) \\ &= s \bigvee_{\chi<\xi} s^*Q_\chi \\ &= \bar{Q}_\xi \end{aligned}$$

It follows that

$$\begin{aligned} Q^s_\xi &= Q^s \bar{Q}^s_\xi \\ &= s^* Q(s\bar{Q}^s_\xi) \\ &= s^* Q(\bar{Q}_\xi) \\ &= s^* Q_\xi \end{aligned}$$

Now let Q have length α. Then for any ξ

$$\begin{aligned} Q^s_\xi \leq \bar{Q}^s_\xi \ &\Rightarrow \ s^* Q_\xi \leq \bar{Q}^s_\xi \\ &\Rightarrow \ Q_\xi \leq s\bar{Q}^s_\xi = \bar{Q}_\xi \\ &\Rightarrow \ \xi \geq \alpha \end{aligned}$$

and $s\bar{Q}^s_\alpha = \bar{Q}_\alpha = 1$, whence $\bar{Q}^s_\alpha = 1$. Thus Q^s has length α.

4.4.10 DEFINITION. *A P-pair satisfying (S) will be called smooth for Q.*

4.4.11 PROPOSITION. *If s, s^* are monotonic and $s^*\,;\,s$ is the identity, then s is smooth for any Q (just smooth).*

PROOF: Let $Y' \subseteq Y$. Then

$$\begin{aligned} \bigvee Y' &= ss^* \bigvee Y' \\ &\geq s\left(\bigvee_{y\in Y'} s^* y\right) \\ &\geq \bigvee_{y\in Y'} ss^* y \\ &= \bigvee Y' \end{aligned}$$

which implies

$$s\left(\bigvee_{y\in Y'} s^* y\right) = \bigvee Y'$$

So

$$s \bigvee_{\chi<\xi} s^* Q_\chi = \bigvee_{\chi<\xi} Q_\chi$$

$$= \tilde{Q}_\xi$$

∎

4.5 Inductions Created by Functors

In this section we look at how P-pairs and P-families can be associated with certain kinds of functor, thereby enabling inductions to be created on image-objects.

4.5.1 A Linear Functor.

Let $F : C \to C'$ be a linear functor. Given $x \in \mathrm{obj}C$, define the function

$$\forall : \mathrm{P}(Fx) \to \mathrm{P}x : q \mapsto \bigvee\{p \mid Fp \leq q\}$$

Then $Fp \leq q \Rightarrow p \leq \forall q$, and $\forall$ is clearly isolating, so

$$\forall : \mathrm{P}(Fx) \leftrightarrows \mathrm{P}x : F$$

is an isolating P-pair. By linearity $F(\forall q) \leq q$, so $\{p \mid Fp \leq q\} = \langle\forall q]$. If F is disjunctive, $F(\forall q) = q$ and we have a bi-P-pair.

If F is bimonotonic, $\forall(Fp) = p$ with $\forall$ monotonic, so $\mathrm{P}(Fx)$ projects onto Fx (qua Type2 objects). In this case, therefore, the P-pair $\langle\forall, F\rangle$ is smooth, and every induction on x creates one of the same length on Fx (if F is also disjunctive, we get order isomorphism and a fortiori a P-iso).

4.5.2 A Monolinear Bifunctor.

Let $! : C^2 \to C'$ be a *mono*linear bifunctor. Choose objects $x, y \in C$, and consider the P-family of 4.3.3(1) applied to the orders $\mathrm{P}x, \mathrm{P}y$, thus:

$$s : \mathrm{P}x \times \mathrm{P}y \to \mathrm{P}x : \langle p, q\rangle \mapsto \text{if } q = 1 \text{ then } p \text{ else } 0$$

and for $p \in \mathrm{P}x$

$$s_p : \mathrm{P}x \times \mathrm{P}y \to \mathrm{P}y : \langle p', q' \rangle \mapsto \text{if } p' \geq p \text{ then } q' \text{ else } 0$$

$$s_p^* : \mathrm{P}y \to \mathrm{P}x \times \mathrm{P}y : q \mapsto \langle p, q \rangle$$

We can now compose the P-pair $\dot\forall : \mathrm{P}(x \,!\, y) \leftrightarrows \mathrm{P}x \times \mathrm{P}y : !$ from (4.5.1) with this P-family to obtain a P-family from $\mathrm{P}(x \,!\, y)$ to $\mathrm{P}x, \mathrm{P}y$. For $p \in \mathrm{P}x,\ q \in \mathrm{P}y$ define the functions

$$\forall_q^x : \mathrm{P}(x \,!\, y) \to \mathrm{P}x : r \mapsto \bigvee\{p' \mid p' \,!\, q \leq r\}$$

$$\forall_p^y : \mathrm{P}(x \,!\, y) \to \mathrm{P}y : r \mapsto \bigvee\{q' \mid p \,!\, q' \leq r\}$$

Then $s(\forall r) = \forall_y^x r$ and $s_p(\forall r) = \forall_p^y r$, so

$$\{\forall_p^y : \mathrm{P}(x \,!\, y) \leftrightarrows \mathrm{P}y : p \,!\, (\cdot) \mid p \in \mathrm{P}x\}$$

is a P-family for $\forall_y^x : \mathrm{P}(x \,!\, y) \to \mathrm{P}x$.

We know from 4.3.3(1) that Q^A, Q^B of lengths α, β respectively create a Q of length $\beta + \alpha$ on $\mathrm{P}x \times \mathrm{P}y$, so if F is bimonotonic Q creates an induction of the same length on $\mathrm{P}(x \,!\, y)$. Thus ! behaves like a sum, although it has no injections, in that every property on $x \,!\, y$ in effect splits into a union of one on x and one on y.

4.5.3 A Bilinear Bifunctor.

Let $\otimes : C^2 \to C'$ be a *bi*linear bifunctor. Once again we can define $\forall_q^x,\ \forall_p^y$, but they no longer factor through $\forall$ as in (4.5.2), because $\{\langle p, q \rangle \mid p \otimes q \leq r\}$ is no longer a *rectangle* — the product of its component projections. However, they do factor through the P-family of 4.3.3(2) over $\mathrm{P}x \wr \mathrm{P}y$, as we now show.

Define $s : \mathrm{P}((x \otimes y) \leftrightarrows \mathrm{P}x \wr \mathrm{P}y : s^*$ by

$$sr = \lambda p.\forall_p^y r$$

$$s^*f = \bigvee_{p\leq x} (p \otimes fp)$$

Then

$$\begin{aligned} s^*f \leq r &\Rightarrow p \otimes fp \leq r \quad \forall p \in \mathrm{P}x \\ &\Rightarrow fp \leq \forall^y_p r \quad \forall p \in \mathrm{P}x \\ &\Rightarrow f \leq sr \end{aligned}$$

and

$$\begin{aligned} sr = 1 &\Rightarrow \forall^y_x r = y \\ &\Rightarrow x \otimes y = x \otimes \forall^y_x r = \bigvee\{x \otimes q \mid x \otimes q \leq r\} \leq r \\ &\Rightarrow r = 1 \end{aligned}$$

Hence $\langle s, s^*\rangle$ is an isolating P-pair, and creates inductions on $x \otimes y$ from those on $\mathrm{P}x\wr\mathrm{P}y$. Now, given inductions Q^x, Q^y on x, y respectively, let Q be the lexicographic induction created by the Galois P-family on $\mathrm{P}x \wr \mathrm{P}y$. We would like to know under what conditions s will be smooth for Q, and create a full lexicographic induction on $x \otimes y$.

Let Q^x, Q^y have lengths α, β repectively, so that Q has length $\beta\alpha$. Recall that, for $\xi < \alpha,\ \chi < \beta$

$$\begin{aligned} Q_{\beta\xi+\chi} &= [Q^x_\xi \to Q^y_\chi] \\ \bar{Q}_{\beta\xi+\chi} &= [\bar{Q}^x_\xi \to 1, \bar{Q}^x_{\xi+1} \to \bar{Q}^y_\chi] \end{aligned}$$

Now

$$\begin{aligned} s^*[p \to q] &= \bigvee_{p'\leq p} (p' \otimes q) \vee \bigvee_{p'\not\leq p} (p' \otimes 0) \\ &= p \otimes q \end{aligned}$$

So

$$\begin{aligned} s\left(\bigvee_{\eta<\beta\xi+\chi} (s^* Q_\eta)\right) &= s\left(\bigvee_{\xi'<\xi} (Q^x_{\xi'} \otimes y) \vee (Q^x_\xi \otimes \bar{Q}^y_\chi)\right) \\ &= s\left((\bar{Q}^x_\xi \otimes y) \vee (Q^x_\xi \otimes \bar{Q}^y_\chi)\right) \\ (\dagger) \qquad &= \lambda p.\left(\text{largest } q \text{ such that } p \otimes q \leq (\bar{Q}^x_\xi \otimes y) \vee (Q^x_\xi \otimes \bar{Q}^y_\chi)\right) \end{aligned}$$

We now need to be able to acquire some information about a rectangle $(p \otimes q)$ contained in a sup of rectangles. We can picture the situation as shown in Diagram 4.1. If we seek inspiration from Cartesian product in Set, we find that it satisfies the following *rectangle property*

$$0 < p \otimes q \leq (p_1 \otimes q_1) \vee (p_2 \otimes q_2) \;\Rightarrow\; \left\{ \begin{array}{ll} p \leq p_1 \;\&\; q \leq q_1 & \text{or} \\ p \leq p_2 \;\&\; q \leq q_2 & \text{or} \\ p \leq p_1 \wedge p_2 & \text{or} \\ q \leq q_1 \wedge q_2 & \end{array} \right\}$$

Let us therefore assume this property for $\otimes$ (notice that it implies definiteness). Then $p \otimes q \leq (\bar{Q}^x_\xi \otimes y) \vee (Q^x_\xi \otimes \bar{Q}^y_\chi)$ implies $p \leq \bar{Q}^x_\xi$ or $p \leq Q^x_\xi \;\&\; q \leq \bar{Q}^y_\chi$, which implies in turn that $p \leq \bar{Q}^x_\xi$ or $q \leq \bar{Q}^y_\chi$, so that we can continue from (†) with

$$\begin{aligned} &= \lambda p.\ \text{if } p \leq \bar{Q}^x_\xi \text{ then } y \text{ elseif } p \leq \bar{Q}^x_{\xi+1} \text{ then } \bar{Q}^y_\chi \text{ else } 0 \\ &= [\bar{Q}^x_\xi \to y,\ \bar{Q}^x_{\xi+1} \to \bar{Q}^y_\chi] \\ &= \bar{Q}_{\beta\xi+\chi} \end{aligned}$$

Hence s is smooth if $\otimes$ is rectangular.

Suppose now that $\otimes$ actually is Cartesian product $(\times)$ on Set (or more generally that C is Set-like and $\otimes$ lifts $\times$). Let Q^x, Q^y be singleton, and for $\eta \in \text{Ord}$ let $Q^x_\eta \setminus \bar{Q}^x_\eta = \{x_\eta\}$, likewise Q^y. Then we have

$$\begin{aligned} Q_{\beta\xi+\chi} \setminus \bar{Q}_{\beta\xi+\chi} &= (Q^x_\xi \otimes Q^y_\chi) \setminus ((\bar{Q}^x_\xi \otimes y) \cup (Q^x_\xi \otimes \bar{Q}^y_\chi)) \\ &= ((Q^x_\xi \setminus \bar{Q}^x_\xi) \otimes Q^y_\chi) \cap (Q^x_\xi \otimes (Q^y_\chi \setminus \bar{Q}^y_\chi)) \\ &= \{x_\xi\} \otimes \{y_\chi\} \\ &= \{\langle x_\xi, y_\chi \rangle\} \end{aligned}$$

So Q is singleton and generates the lexicographic order on $x \otimes y$. This case justifies the description of nested induction as "induction on y within x".

Induction on Joins and Sums.

We now see how a join in bicontext C defines a P-pair. We shall make use of 3.5.3. Let $+$ be a join of arity I and let $x \in \mathrm{obj}C^I$.

4.5.4 THEOREM. *$(\cdot)| : +x \leftrightarrows x : +$ is a P-pair, isolating if $+$ is a sum, and a bi-P-pair if $+$ is disjunctive.*

PROOF: Let $p \leq x$, $q \leq +x$. Then $+p \leq q \Rightarrow p \leq (+p)| \leq q|$ so we have a P-pair. If $+$ is a sum, then $q| = 1 \Rightarrow q \geq +(q|) = 1$ so the P-pair is isolating.

Now suppose $+$ is disjunctive. Then $q| \leq p \Rightarrow q \leq +(q|) \leq +p$, making $\langle +, (\cdot)| \rangle$ a P-pair. ∎

4.5.5 THEOREM. *$(\cdot)|i : x_i \leftrightarrows +x : (\cdot)\_i$ is a P-pair.*

PROOF: $p\_i \leq q \Rightarrow p = p\_i|i \leq q|i$ ∎

We can now tie this in with the sum-like behaviour of a (mono)linear bimonotonic functor.

4.5.6 THEOREM. *If $+$ is a sum, $(\cdot)| = \forall$.*

PROOF: Let $q \in \mathrm{P}(+x)$. Then $q|$ is the largest $p \in \mathrm{P}x$ such that $+p \leq q = \bigvee\{p \in \mathrm{P}x \mid +p \leq q\} = \forall q$. ∎

4.6 PRODUCTS

We pointed out earlier that it does not seem possible to find a satisfactory limit notion of *product* in a bicontext. But for a locally-complete bicontext we have just seen that a bilinear rectangular bifunctor behaves very much like a product as far as induction is concerned. Certainly ordinary Cartesian product in Set (and other Set-like bicontexts) has these properties (plus exactness).

In contrast, the monolinearity of Cartesian product in Type2 makes this functor create inductions via $\forall$ in the same way that it does via its injections qua sum.

It may be wisest not to try to establish a general notion of product, but if we do, the best candidates are probably the polylinear rectangular functors, possibly also disjunctive (the rectangular property needs extension to higher arities, and becomes rather unpleasant — we shall keep to the dyadic case here).

Let us therefore define

4.6.1 DEFINITION. *Let* $\otimes : C^2 \to C$ *be exact. If it is bimonotonic monolinear, it is* a weak product; *if it is rectangular bilinear, it is* a strong product.

As with joins and sums, one would want to be able to lift products via a forgetful functor. We already know that the above properties, bar rectangularity, do so (disjunctivity requires the composite of forgetful functor followed by underlying functor to be disjunctive). We now show that rectangularity also lifts.

4.6.2 THEOREM. *If functor* $\otimes : C^2 \to C$ *lifts* $\otimes' : C'^2 \to C'$ *via forgetful* $V : C \to C'$, *then* $\otimes$ *is rectangular if* $\otimes'$ *is.*

PROOF: Let f, g be arrows in C^2. Then

$$0 < f \otimes g \leq (f_1 \otimes g_1) \vee (f_2 \otimes g_2) \Rightarrow \begin{cases} 0 < V(f \otimes g) \\ \quad = Vf \otimes' Vg \\ \quad \leq V(f_1 \otimes g_1) \vee V(f_2 \otimes g_2) \\ \quad = (Vf_1 \otimes' Vg_1) \vee (Vf_2 \otimes' Vg_2) \end{cases}$$

$$\Rightarrow \begin{cases} Vf \leq Vf_1 \wedge Vf_2 \quad \text{or} \\ Vg \leq Vg_1 \wedge Vg_2 \quad \text{or} \\ Vf \leq Vf_1 \ \& \ Vg \leq Vg_1 \quad \text{or} \\ Vf \leq Vf_2 \ \& \ Vg \leq Vg_2 \end{cases}$$

$$\Rightarrow \begin{cases} f \leq f_1 \wedge f_2 \quad \text{or} \\ g \leq g_1 \wedge g_2 \quad \text{or} \\ f \leq f_1 \ \& \ g \leq g_1 \quad \text{or} \\ f \leq f_2 \ \& \ g \leq g_2 \end{cases}$$

∎

4.6.3 COROLLARY. *$\otimes$ is a weak, resp. strong product if $\otimes'$ is.*

Examples.

(1) Cartesian product in Set, as we have already seen is a disjunctive strong product.

(2) Coalesced product ($\oplus$) in Set_0, defined as the canonical lift of the Cartesian product in Set (via *drop*) with

$$x \oplus y = (drop\ x \times drop\ y) \cup \{0\}$$

is therefore a disjunctive strong product.

(3) In Simple2 (Type2|Simp) Cartesian product is a (non-disjunctive) weak product.

The following property of strong products makes them even more appealing.

4.6.4 THEOREM. *If C is locally-complete and $\otimes : C^2 \to C$ is definite and disjunctive, the inf of a non-empty set of rectangles is a rectangle.*

PROOF: Let $A \subseteq \mathrm{P}x$, $B \subseteq \mathrm{P}y$ for $x, y \in \mathrm{obj}C$. Let $p = \bigwedge A$, $q = \bigwedge B$. If $\bigwedge(A \otimes B) = 0$ the result is trivial, so assume not. Then $p \otimes q \leq A \otimes B$, and for any $p' \leq x, q' \leq y$

$$\begin{aligned} 0 < p' \otimes q' \leq A \otimes B \;&\Rightarrow\; p' \leq p \;\&\; q' \leq q \\ &\Rightarrow\; p' \otimes q' \leq p \otimes q \end{aligned}$$

So $p \otimes q$ is the greatest rectangular lower bound of $A \otimes B$. But by disjunctivity, every lower bound of $A \otimes B$ is a sup of such, hence $p \otimes q = \bigwedge(A \otimes B)$. ∎

Chapter 5

Algebraic Types

In this chapter we construct and investigate a bicontext of algebraic types. It is a Set-like bicontext, in contrast to the more usual sub of **Type2**. In Chapter 6, we shall look at the relationship between the two.

Convention.

We shall use square notation for the orderings *within* the types and for the pointwise ordering on functions between types, reserving the pointed notation for the bundle orderings of the bicontext. Also, the term "isomorphism" will mean "$\sqsubseteq$-isomorphism".

5.1 Alg

Let *cent* be the function over the class Alg of algebraic types that carries each one to its centre. Then

5.1.1 DEFINITION. $\mathbf{Alg} = \mathbf{Set} \times cent$.

Thus Alg = obj(**Alg**). We can obviously factor *cent* through $drop : \mathbf{Set} \to \mathbf{Set}_0$, so **Alg** is also $\mathbf{Set}_0$-like.

We now define some spanning subs of **Alg**, having first introduced the idea of a generator.

5.1.2 DEFINITION. *Let* $x, y \in$ Alg *with* $r \subseteq x| \times y|$. r *is a* generator *when it is monotonic and* $!r$ *is* $\mathcal{U}$*-closed* .

The next definition and theorem justify the term generator.

5.1.3 DEFINITION. *For generator $r \subseteq x| \times y|$, define the function $[r] : x \to y$ (acting on all of x), by*

$$[r]x = \bigsqcup r(x|!r)$$

This is a good definition because $!r$, being $\mathcal{U}$-closed, is projective, so $(x|!r)$ is directed or empty, whence $r(x|!r)$ is also.

Henceforth we shall write $x|r$ for $x|!r$.

5.1.4 THEOREM. *$[r]$ is the smallest (with respect to $\sqsubseteq$) map extending r. If $0 \notin !r$, $[r]$ is strict.*

PROOF:

(a)

$$\begin{aligned} x \sqsubseteq x' &\Rightarrow x|r \subseteq x'|r \\ &\Rightarrow r(x|r) \subseteq r(x'|r) \\ &\Rightarrow [r]x \sqsubseteq [r]x' \end{aligned}$$

so $[r]$ is monotonic.

(b)

$$\begin{aligned} x|r = \bigcup_{a \in x|} (a|r) &\Rightarrow r(x|r) = \bigcup_{a \in x|} r(a|r) \\ &\Rightarrow [r]x = \bigsqcup_{a \in x|} [r]a \end{aligned}$$

so $[r]$ is continuous.

(c) Let $a \in !r$. Then a is top in $x|r$, ra is top in $r(x|r)$, so $[r]a = ra$, whence $r \subseteq [r]$. Now let $r \subseteq f$, a map $: x \to y$. Then for $a \in x|r$, $ra = fa \sqsubseteq fx$. Thus $r(x|r) \sqsubseteq fx$, so $[r]x \sqsubseteq fx$ and $[r]$ is the smallest extension.

Finally, if $0 \notin !r$, $0|r = \emptyset$, giving $[r]0 = 0$. ∎

We now examine some interesting bisubs of Alg.

$\mathrm{Alg}_{\sqsubseteq}$.

This is the class of of bimonotonic relations between members of Alg. Clearly it is a bisub (it is D-closed because each relation is determined by its finite sub-relations). It is also hereditary, almost-closed, and a sub of Int(Alg). We shall call its arrows $\sqsubseteq$-*links*, and usually denote them by u, v, etc. Notice that such a u is an isomorphism between its left- and right-sets, and that the left-(right-)end of u is the identity on its left-(right-)set. We may therefore think of the latter as *being* the ends of u if no confusion will arise.

$\mathrm{Alg}_{\vdash}$.

This is the class of $r : x \rightarrow y \in$ Alg such that, if $R, R' \subseteq_{\mathrm{fin}} r$ then $!R \vdash !R'$ iff $R! \vdash R'!$. Again, it is an hereditary bisub of Alg. We shall refer to its members as $\vdash$-*links*.

5.1.5 PROPOSITION. $\mathrm{Alg}_{\vdash} \subseteq \mathrm{Alg}_{\sqsubseteq}$

PROOF: $(\forall x, x')\ x \sqsubseteq x' \Leftrightarrow x' \vdash x$. Thus, taking $R = \{\langle x, y\rangle\}, R' = \{\langle x', y'\rangle\}$, $x \sqsubseteq x' \Leftrightarrow y \sqsubseteq y'$. ∎

$\mathrm{Alg}_{\mathcal{U}}$.

This is the class of $r : x \rightarrow y \in$ Alg such that, if $R \subseteq_{\mathrm{fin}} r$, then $!R$ has a finite $\mathcal{U}$-closure iff $R!$ does, and when they do $R \subseteq \exists\theta : \mathcal{U}^*(!R) \cong \mathcal{U}^*(R!)$. We may refer to its members as $\mathcal{U}$-*links*.

5.1.6 THEOREM. *$\mathrm{Alg}_{\mathcal{U}}$ is an hereditary bisub.*

PROOF: We show closure under composition and D-closure. The other conditions are obvious.

(;) Let $R \subseteq_{\mathrm{fin}} r\,;s$. Choose y for each $\langle x, z\rangle \in R$ such that $x\ r\ y\ s\ z$. Let $R_1 = \{\langle x, y\rangle\}, R_2 = \{\langle y, z\rangle\}$. Then $!R = !R_1$, $R_1! = !R_2$, $R_2! = R!$, so $!R$ has a finite $\mathcal{U}$-closure iff $R!$ does. When they do, let isomorphisms $\theta_1 \supseteq R_1, \theta_2 \supseteq R_2$. Then $R \subseteq R_1\,; R_2 \subseteq \theta_1\,; \theta_2$.

(D) Let $D \nearrow r$, and let $R \subseteq_{\text{fin}} r$. Choose $d \in D$ such that $R \subseteq d$. Then $!R$ has a finite $\mathcal{U}$-closure iff $R!$ does. If they do, there is an isomorphism extending R by virtue of $R \subseteq d$. ∎

5.1.7 PROPOSITION. $\mathrm{Alg}_{\mathcal{U}} \subseteq \mathrm{Alg}_{\sqsubseteq}$.

PROOF: Let $R = \{\langle x, y\rangle, \langle x', y'\rangle\} \subseteq r \in \mathrm{Alg}_{\mathcal{U}}$, with $x \sqsubseteq x'$. Then $\mathcal{U}^*(!R) = \{x, x'\}$, so if θ is the extending isomorphism, $R \subseteq \theta$. Thus $y \sqsubseteq y'$. ∎

Observe that each $\mathrm{Alg}_{\vdash,\mathcal{U}}$ is abstract in $\mathrm{Alg}_{\sqsubseteq}$.

Can either of the inclusions we have just proved be reversed? We present counter-examples to show that neither of $\mathrm{Alg}_{\vdash}, \mathrm{Alg}_{\mathcal{U}}$ is contained in the other, whence it follows that neither of the inclusions is reversible.

(1) $\mathrm{Alg}_{\mathcal{U}} \not\subseteq \mathrm{Alg}_{\vdash}$: Let

$$x^\circ = \omega \cup \{a, b, c\} \text{ with } \{a, b, c\} \sqsubseteq \omega$$

$$y^\circ = x^\circ \text{ except that } c \not\sqsubseteq 0$$

— see Diagram 5.1. Let $u \subseteq x^\circ \times y^\circ = \{\langle a, a\rangle, \langle b, b\rangle, \langle c, c\rangle\}$. Then u is a $\mathcal{U}$-link because only the singleton subsets of $\{a, b, c\}$ have a $\mathcal{U}$-closure (which is the subset itself). But $\{a, b\} \vdash c$ in x, yet not in y, so u is not a $\vdash$-link.

(2) $\mathrm{Alg}_{\vdash} \not\subseteq \mathrm{Alg}_{\mathcal{U}}$: Let

$$x^\circ = \{a, b, 0, 1\} \text{ with } \{a, b\} \sqsubseteq \{0, 1\}$$

$$y^\circ = \{a, b, 0, 1, 2\} \text{ with } \{a, b\} \sqsubseteq \{0, 1, 2\}$$

— see Diagram 5.2. Let $u \subseteq x^\circ \times y^\circ = \{\langle a, a\rangle, \langle b, b\rangle, \langle 0, 0\rangle\}$. Then the only entailments with irreducible RHS that hold amongst $a, b, 0$ on either side are $a \vdash a$, $b \vdash b$, $c \vdash c$, $0 \vdash 0$, $0 \vdash a$, $0 \vdash b$. So u is a $\vdash$-link. But $\mathcal{U}^*\{a, b\}$ is the whole centre on either side, so they cannot be isomorphic. Thus u is not a $\mathcal{U}$-link.

Nevertheless, there is an interesting common subclass of all three subs, which we define below. First, we prove some useful lemmata.

5.1.8 LEMMA. *Let $y \subseteq x \in \mathrm{Alg}$ be $\mathcal{U}$-closed. Then the inclusion is in $\mathrm{Alg}_{\vdash} \cap \mathrm{Alg}_{\mathcal{U}}$*

PROOF:

($\vdash$) Let $Y, Y' \subseteq_{\mathrm{fin}} y$. Clearly $Y \vdash_x Y' \Rightarrow Y \vdash_y Y'$ (the subscript on $\vdash$ indicates the order it refers to). Suppose, conversely, that $Y \vdash_y Y'$, and let some $x \sqsupseteq Y$. Then $\exists y \in \mathcal{U}Y$ ($\subseteq y$) with $x \sqsupseteq y \sqsupseteq Y$. Thus $x \sqsupseteq y \sqsupseteq \exists y' \in Y'$, giving $Y \vdash_x Y'$.

($\mathcal{U}$) Let $B \subseteq_{\mathrm{fin}} y$. Clearly $\mathcal{U}_x B$ ($\subseteq y$) is a y-roof of B, so $\mathcal{U}_y B$ exists $\subseteq \mathcal{U}_x B$, so every $\mathcal{U}_x$-closed $Y \subseteq y$ is $\mathcal{U}_y$-closed (in particular, y itself). But then $a \sqsupseteq b \sqsupseteq B$, so b is an upper bound of B, whence $b = a$. Therefore $\mathcal{U}_x B = \mathcal{U}_y B$.

It follows that $\mathcal{U}_y^* B$ is $\mathcal{U}_x$-closed, whence $\mathcal{U}_x^* B = \mathcal{U}_y^* B$, which yields the result.

∎

5.1.9 COROLLARY.

(1) *If x is SFP, so is y.*

(2) *The inclusion : $x^\circ \subseteq x$ is a $\mathcal{U}$-link, so it is immaterial whether $\vdash, \mathcal{U}$ are taken relative to x° or x.*

(3) *If $u : x \to u \in \mathrm{Alg}_{\sqsubseteq}$ has $!u, u!$ both $\mathcal{U}$-closed, it is in $\mathrm{Alg}_{\vdash} \cap \mathrm{Alg}_{\mathcal{U}}$.*

PROOF: (1) and (2) are immediate, and (3) follows by observing that u is the three-fold composite

$$(x \supseteq\, !u) \,;\, (u : !u \cong u!) \,;\, (u! \subseteq y)$$

of relations in both subs. ∎

5.1.10 LEMMA. *If $u : x \to y \in \mathrm{Alg}_{\vdash}$ with $!u$ $\mathcal{U}$-closed, $u!$ is also $\mathcal{U}$-closed, whence $u \in \mathrm{Alg}_{\mathcal{U}}$.*

PROOF: Let $u(X) \subseteq_{\mathrm{fin}} u!$. Then $X \vdash \mathcal{U}X \subseteq\, !u$, so $u(X) \vdash u(\mathcal{U}X)$. Also, $\forall x \in X, x' \in \mathcal{U}X$ we have $x' \vdash x$ giving $ux' \vdash ux$. Thus $u(\mathcal{U}X)$ is a roof of $u(X)$, whence $u!$ is $\mathcal{U}$-closed. ∎

It follows that all three kinds of links with $\mathcal{U}$-closed ends coincide; and a $\vdash$-link with one $\mathcal{U}$-closed end already has both ends $\mathcal{U}$-closed. So define

5.1.11 DEFINITION. Alg_* *is the class of* $\sqsubseteq$*-links with* $\mathcal{U}$*-closed ends.*

5.1.12 THEOREM. $\mathrm{Alg}_* \triangleleft^= \mathrm{Alg}_{\sqsubseteq,\vdash,\mathcal{U}}$

PROOF: Both the 0-closure and the symmetry are obvious. And since $\mathcal{U}^*$ is algebraic, Alg_* is D-closed. If $u : x \to y, v : y \to z$ are in Alg_*, $!(u;v) = !(u\,;(u! \cap !v))$, which is $\mathcal{U}$-closed because $(u\,;(u! \cap !v))! = u! \cap !v$ is (using Lemma 5.1.10). ∎

5.2 SFP AND SIMPLE TYPES

If SFP and Simple are the classes of SFP and simple types respectively, and if ? stands for any of the suffices introduced in the previous section, then we define

5.2.1 DEFINITION.

(1) $\mathrm{SFP}_? = \mathrm{Alg}_?|\mathrm{SFP}$

(2) $\mathrm{Simp}_? = \mathrm{Alg}_?|\mathrm{Simp}$

Referring back to the counter-examples for $\mathrm{Alg}_{\mathcal{U}} \not\subseteq \mathrm{Alg}_{\vdash}$ and vice versa, the types used in the first were not SFP, whereas those in the second were SFP but not simple. So the second also proves that $\mathrm{SFP}_{\vdash} \not\subseteq \mathrm{SFP}_{\mathcal{U}}$; but beyond this the counter-examples cannot be strengthed, as the next theorem shows.

5.2.2 THEOREM.

(1) $\mathrm{SFP}_{\mathcal{U}} \subseteq \mathrm{SFP}_{\vdash}$

(2) $\mathrm{Simp}_{\mathcal{U}} = \mathrm{Simp}_{\vdash}$

PROOF:

(1) Let $u : x \to y \in \mathrm{SFP}_{\mathcal{U}}$. Let $R, R' \subseteq_{\mathrm{fin}} u$ with $!R \vdash !R'$. Extend $R \cup R'$ to an isomorphism between $\mathcal{U}^*(!R \cup !R')$ and $\mathcal{U}^*(R! \cup R'!)$. Then the entailment

$!R \vdash !R'$ holds relative to $\mathcal{U}^*(!R \cup !R')$, so $R! \vdash R'!$ holds in $\mathcal{U}^*(R! \cup R'!)$, hence in y. Symmetry gives the result.

(2) First of all, note that if x is simple and $X \vdash X'$ $(X, X' \subseteq_{\text{fin}} x)$, then either X is inconsistent, in which case $X \vdash \emptyset$, or $\bigsqcup X \sqsupseteq \exists x' \in X'$, when $X \vdash x'$. Thus we need only consider entailments with zero or one elements on the RHS, expressing inconsistency and RHS $\sqsubseteq \bigsqcup$ LHS respectively.

By virtue of (1), we need only prove $\mathrm{Simp}_{\vdash} \subseteq \mathrm{Simp}_{\mathcal{U}}$. But $\mathcal{U}^*$ is dual-closure in the centre of a simple type. And if $x_1, \ldots, X_n, X \subseteq_{\text{fin}} x$ then $\{\bigsqcup X_i \mid i = 1, ..., n\} \vdash \bigsqcup X$ (or $\emptyset$) iff $\bigcup_{i=1}^n X_i \vdash \forall x \in X$ (or $\emptyset$). So every entailment in $\mathcal{U}^* X$ is equivalent to an entailment on X.

So let $u : x \to y \in \mathrm{Simp}_{\vdash}$ and let $R \subseteq_{\text{fin}} u$. Then $\mathcal{U}^*!R$ is determined (up to isomorphism) by all its entailments, and therefore by all the entailments within $!R$. But these are identical, modulo u, to those on $R!$. Hence u extends to an isomorphism between $\mathcal{U}^*!R$ and $\mathcal{U}^*R!$. ∎

Of course, these extra inclusions invalidate, for SFP, our original proof (from the two counter-examples) that $\sqsubseteq \not\subseteq \vdash$. However, even in Simp this still holds. For let

$$x^\circ = \{0, 1, \top\} \text{ with } \{0, 1\} \sqsubseteq \top$$

$$y^\circ = \{0, 1, a, \top\} \text{ with } \{0, 1\} \sqsubseteq a \sqsubseteq \top$$

— see Diagram 5.3. And let $u \subseteq x^\circ \times y^\circ = \{\langle 0, 0\rangle, \langle 1, 1\rangle, \langle \top, \top\rangle\}$. Clearly u is a $\sqsubseteq$-link, yet $\{0, 1\} \vdash \top$ in x but not in y.

5.3 Colimits

In this section, we shall use some arbitrary but fixed directed set N as a net. If $m \leq n$, we shall write m, n (as in Chapter 3) for the unique arrow from $: m \to n$. All diagrams, cones etc. will be over N.

5.3.1 THEOREM. *Any injective diagram in* $\mathrm{Alg}_{\sqsubseteq}$ *has a colimit.*

PROOF: This is a standard construction. Let $[\![\cdot]\!]$ be the injective diagram, and construct the algebraic type generated by X under preorder $\preceq$ factored by its corresponding equivalence $\simeq$, where:

(a) X is the disjoint union of all the $[\![n]\!]$ $(n \in N)$. We shall write x_n for the typical element of $[\![n]\!]$.

(b) $\preceq$ is the preorder on X defined by:

$$x_m \preceq x_n \Leftrightarrow m, n \leq \exists p \text{ with } [\![m,p]\!]x_m \sqsubseteq [\![n,p]\!]x_n$$

$\preceq$ is obviously relexive. To see transitivity, we first show that the choice of p is immaterial. For if $m, n \leq p'$, let $p, p' \leq p''$. Then

$$\begin{aligned}
[\![m,p']\!]x_m &= [\![p',p'']\!]^-([\![m,p'']\!]x_m) \quad \text{since } [\![p',p'']\!]^L = 1 \\
&= [\![p',p'']\!]^-([\![p,p'']\!]([\![m,p]\!]x_m)) \\
&\sqsubseteq [\![p',p'']\!]^-([\![p,p'']\!]([\![n,p]\!]x_n)) \\
&= [\![p',p'']\!]^-([\![n,p'']\!]x_n) \\
&= [\![n,p']\!]x_n
\end{aligned}$$

So if $x_m \preceq x_n \preceq x_p$, let $q \geq m, n, p$. Then

$$[\![m,q]\!]x_m \sqsubseteq [\![n,q]\!]x_n \sqsubseteq [\![p,q]\!]x_p$$

Now let c_n be the canonical projection : $[\![n]\!] \to (X/\simeq)^\circ = c_\infty$. Then

(i) $x_n \sqsubseteq x'_n \Rightarrow x_n \preceq x'_n$, so c_n is monotonic

(ii)
$$\begin{aligned}
x_n \preceq x'_n &\Rightarrow [\![n,p]\!]x_n \sqsubseteq [\![n,p]\!]x'_n \ \exists p \geq n \\
&\Rightarrow x_n \sqsubseteq x'_n
\end{aligned}$$

because $[\![n,p]\!]$ is bimonotonic. So c_n is bimonotonic; since it is also total, it is an injective $\sqsubseteq$-link.

(iii) For $m \leq n$, $x_m \simeq [\![m,n]\!]x_m$, whence $c_m x_m = c_n([\![m,n]\!]x_m)$.

Thus c is a cone; it remains to show that it is unit. But c_n^R is the set of $\simeq$-classes that intersect $[\![n]\!]$, so that $x = \bigcup_n c_n^R$. ∎

The next theorem says that this colimit construction stays inside any of the $(\cdot)_\vdash$ subs.

5.3.2 THEOREM. *If the diagram of the previous theorem is in $\Gamma_\vdash$, so is the colimit, where Γ can be any of* Alg, SFP *or* Simp.

PROOF:

(1) $\mathrm{Alg}_\vdash$: Let $A_n, B_n \subseteq_{\mathrm{fin}} [\![n]\!]$. First suppose that $c_n A_n \vdash c_n B_n$. Then if $y_n \in [\![n]\!]$ is an upper bound of A_n, $c_n y_n \sqsupseteq c_n A_n$, so $c_n y_n \sqsupseteq c_n b_n$, $\exists b_n \in B_n$. Thus $y_n \sqsupseteq b_n$, whence $A_n \vdash B_n$.

Conversely, let $A_n \vdash B_n$, and let $x \in c_\infty$ be an upper bound of $c_n A_n$. Choose $p \geq n$ such that $x = c_p x_p$. Then $[\![n,p]\!]A_n \vdash [\![n,p]\!]B_n$, and $x \sqsupseteq c_p([\![n,p]\!]A_n)$. Hence $\exists b_n \in B_n$, $x \sqsupseteq c_p([\![n,p]\!]B_n) = c_n b_n \in c_n B_n$. Therefore $c_n A_n \vdash c_n B_n$.

(2) $\mathrm{SFP}_\vdash$: We need only show that c_∞ is SFP if every $[\![n]\!]$ is. Since every $[\![m,n]\!]$ is total, the diagram is in SFP_*. Let $A \subseteq_{\mathrm{fin}} c_\infty$. Choose n such that $A = c_n A_n$ with $A_n \subseteq [\![n]\!]$. Then $c_n A_n \subseteq c_n(\mathcal{U}^* A_n)$ which is $\mathcal{U}$-closed and finite.

(3) $\mathrm{Simp}_\vdash$: We need only show that c_∞ is dualmost-complete if every $[\![n]\!]$ is. Let $\{a,b\} \sqsubseteq x \in c_\infty$. Choose n such that $a = c_n a_n$, $b = c_n b_n$ and $x = c_n x_n$ with $a_n, b_n, x_n \in [\![n]\!]$. Then $\{a_n, b_n\} \sqsubseteq x_n$, so $a_n \sqcup b_n$ exists. Now $a_n \sqcup b_n \sqsupseteq \{a_n, b_n\}$ so $c_n(a_n \sqcup b_n) \sqsupseteq \{c_n a_n, c_n b_n\}$, and $\{a_n, b_n\} \vdash a_n \sqcup b_n$, so $\{c_n a_n, c_n b_n\} \vdash c_n(a_n \sqcup b_n)$. Thus $c_n(a_n \sqcup b_n) = c_n a_n \sqcup c_n b_n$. Ie: $a \sqcup b$ exists. ∎

In the second case, c is also in SFP_*. Later we shall see this colimit matching, via an appropriate doublet, the usual construction in Type2.

We have seen that $\vdash$-links are to sufficient to guarantee that colimits preserve simplicity or SFP-hood — why, therefore, introduce $\mathcal{U}$-links? Constructors are the reason; the obvious naming systems for $\underline{(\cdot)}$ (lift), $+$ and $\times$ make them $\mathrm{Alg}_{\vdash}$-constructors, but we are unable to find naming systems for $\rightarrow$ or $\mathbb{P}$ (powertype) to make them $\mathrm{SFP}_{\vdash}$-constructors ($\underline{(\cdot)}, +, \times$ and $\mathbb{P}$ are also $\mathrm{Alg}_{\sqsubseteq}$ constructors). They can all, however, be made into $\mathrm{SFP}_{\mathcal{U}}$-constructors (it is actually only C_F that requires $\mathcal{U}$; C_N gets by with $\vdash$). Of course, they are all $\mathrm{Simp}_{\vdash}$-constructors.

We begin with a useful sufficient condition for C_F. Define

$$\mathbb{F}\mathrm{in} = \mathrm{Alg}|\{\text{finite types}\}$$

($\mathrm{C}_{\mathcal{U}}$) Let $F : \mathrm{Iso}(\mathbb{F}\mathrm{in}^I) \rightarrow \mathrm{Iso}(\mathbb{F}\mathrm{in})$ be a pre-functor such that for any $x \in \mathrm{SFP}^I, X \subseteq x^\circ$ with each X_i $(i \in I)$ finite and $\mathcal{U}$-closed (and therefore itself SFP), there is total bimonotonic $in_X : FX \rightarrow Fx$ with $\mathcal{U}$-closed right-set which satisfies the following:

Let $x, y \in \mathrm{SFP}^I$ with names $s \equiv_m t(x, y)$ satisfying $\mathrm{SFP}_{\mathcal{U}}$, and let A_i, B_i be $\mathcal{U}^*$ of the left- and right-sets respectively of the i^{th} induced relation : $x_i \rightarrow y_i$, with extending isomorphism $\theta : A_i \cong B_i$. Then for each $j = 1, ..., m$

$$\langle [s_j], [t_j] \rangle \in in_A^- \,;\, F\theta \,;\, in_B$$

It follows that if $\mathrm{C}_{\mathcal{U}}$ holds for F, so does C_F, because $in_A^- \,;\, F\theta \,;\, in_B$ is a $\mathcal{U}$-link. If in addition we have

(C_*) Everything in $in_A!$ has a name n with $|n| \subseteq \bigcup_{i \in I} A_i$

then $\mathrm{SFP}_{\mathcal{U}}$-constructor F is within SFP_* (this will be important in Chapter 6). For then, if the tuple s has each $[s_j] \in\, !Fu$ $(u \in \mathrm{SFP}_*)$, $A \leq\, !u$, so $in_A! \subseteq\, !Fu$, whence $!Fu$ is $\mathcal{U}$-closed, ie: $Fu \in \mathrm{SFP}_*$.

We now proceed to show how the common type-transformations do become constructors.

5.4.1 LIFT, $\underline{(\cdot)}$.

This constructor merely adds a new bottom element. It is monadic and for $x \in \mathrm{obj}(\mathrm{SFP})$ is defined by

$$\underline{x}^{\circ} = x^{\circ} \uplus \{0\}$$

It has an obvious immediate definition on finite isomorphisms.

A sort here is just a non-negative integer; define $\mathrm{Op}_0 = \{0\}, \mathrm{Op}_1 = \{\langle\cdot\rangle\}$, and for $n > 1$, $\mathrm{Op}_n = \emptyset$. Let Name = Term and define $[\cdot]$ by

$$[0] = 0, \quad [\langle a \rangle] = a$$

Since $[\cdot]$ is (1,1), $\underline{(\cdot)}$ is trivially directed.

To say that tuples of names $t \equiv_m t'(x, y)$ satisfy $\mathrm{SFP}_{\mathcal{U}}$ is to say that the (components of) the non-zero names match in a $\mathcal{U}$-link. It is then obvious that, if θ is the extending isomorphism between the corresponding $\mathcal{U}$-closures, for each $j = 1, ..., m$

$$[t_j] \, \underline{\theta} \, [t'_j]$$

So $\underline{(\cdot)}$ satisfies $\mathrm{C}_{\mathcal{U}}$. $\mathrm{C}_{\mathcal{N}}$ is immediate. Lastly, everything in the lift of a $\mathcal{U}$-closure of the components of a tuple t has a name whose component is in this $\mathcal{U}$-closure, whence C_* holds.

5.4.2 COALESCED SUM, $+$.

This has arity 2 ($= \{0, 1\}$), and is defined on $x, y \in \mathrm{SFP}$ by

$$(x + y)^{\circ} = x^{\circ} \uplus y^{\circ}$$

It has an obvious immediate definition on finite isomorphisms.

Define Op by $\mathrm{Op}_{\langle 0\rangle} = \{0\}$, $\mathrm{Op}_{\langle 1\rangle} = \{1\}$, all other $\mathrm{Op}_r = \emptyset$, and let Name = Term. Define $[\cdot]$ by

$$[0a] = a \text{ in } (x+y), \quad [1b] = b \text{ in } (x+y)$$

Since $[\cdot]$ is (1,1), + is directed.

Given any finite $\mathcal{U}$-closed $X \subseteq x^\circ, Y \subseteq y^\circ$, the inclusion $in : X + Y \subseteq x + y$ has $\mathcal{U}$-closed right-set.

Now to say that tuples of names $t \equiv_m t'(\langle x, y\rangle, \langle x', y'\rangle)$ satisfy $\mathrm{SFP}_{\mathcal{U}}$ is to say that the arguments of those names of sort 0 match in a $\mathcal{U}$-link : $x \to x'$, likewise those of sort 1. When these are extended to isomorphisms θ_0, θ_1 respectively between the corresponding $\mathcal{U}$-closures, obviously for each $j = 1, ..., m$

$$[t_j]\ (\theta_0 + \theta_1)\ [t'_j]$$

Thus + satisfies $\mathrm{C}_{\mathcal{U}}$. C_N is immediate. And clearly everything in the sum of the $\mathcal{U}$-closures of the components of a tuple t has a name whose components are contained in these $\mathcal{U}$-closures, so C_* holds.

5.4.3 Cartesian Product, $\times$.

Also of arity 2, it has

$$(x \times y)| = x| \times y|$$

Again there is an obvious definition on finite isomorphisms. Define Op by $\mathrm{Op}_{\langle 0\rangle} = \{\langle \cdot, 0\rangle\}$, $\mathrm{Op}_{\langle 1\rangle} = \{\langle 0, \cdot\rangle\}$, $\mathrm{Op}_{\langle 0,1\rangle} = \{\langle \cdot, \cdot\rangle\}$, all other $\mathrm{Op}_r = \emptyset$. Name = Term, and $[\cdot]$ has the obvious definition ($[\langle a, 0\rangle] = \langle a, 0\rangle$,etc.). Again, $[\cdot]$ is (1,1), so $\times$ is directed.

Given any $x, y \in \mathrm{SFP}$ and finite $\mathcal{U}$-closed $X \subseteq x|, Y \subseteq y|$, we show that $X \times Y$ is $\mathcal{U}$-closed, so that again the inclusion has $\mathcal{U}$-closed right-set. But if $V \subseteq X \times Y$, $\mathcal{U}V_0 \times \mathcal{U}V_1$ $(\subseteq X \times Y)$ is a roof of V, because

$$\begin{aligned}\langle x, y\rangle \sqsupseteq V &\Rightarrow x \sqsupseteq V_0 \ \& \ y \sqsupseteq V_1 \\ &\Rightarrow x \sqsupseteq \exists a \in \mathcal{U}V_0 \ \& \ y \sqsupseteq \exists b \in \mathcal{U}V_1\end{aligned}$$

Tuples $t \equiv_m t'(\langle x, y\rangle, \langle x', y'\rangle)$ satisfy $\mathrm{SFP}_{\mathcal{U}}$ when all the first components match in a $\mathcal{U}$-link, and all the second components likewise (zeroes just being omitted).

So if θ_0, θ_1 are the isomorphisms extending the two induced component relations to the $\mathcal{U}$-closures of the tuple components, clearly we have for each $j = 1, ..., m$

$$[t_j]\ (\theta_0 \times \theta_1)\ [t'_j]$$

So again $\mathbf{C}_{\mathcal{U}}$ holds, and $\mathbf{C}_{N}$ is immediate. And again, every $\langle a, b\rangle$ in the product of the $\mathcal{U}$-closures of the components of t has the name $\langle a, b\rangle$, so $\mathbf{C}_*$ holds.

Furthermore, if we have only $\mathrm{Op}_{\langle 0,1\rangle}$ non-empty, the same argument shows that coalesced product, for which

$$(x \oplus y)^\circ = x^\circ \times y^\circ,$$

is also a constructor.

5.4.4 Powertype, $\mathbf{P}$.

This is the (unary) powertype (power-domain) construction of [5]. For $x \in \mathrm{SFP}$, $(\mathbf{P}x)^\circ$ is all the finite subsets of x° under (the quotient order of) the Milner preorder, $\sqsubseteq_{\mathrm{M}}$. Clearly $\mathbf{P}$ has a natural definition on finite isomorphisms. Define Op by, for $n > 0$,

$$\mathrm{Op}_n = \{\langle \ldots\rangle, \langle 0, \ldots\rangle \mid \ldots \text{ represents } n \text{ entries}\}$$

(again a sort is just an integer, $\mathrm{Op}_0 = \emptyset$). Every Term is a Name with $[\cdot]$ defined in the obvious way by

$$[\langle x_1, \ldots, x_n \rangle] = \{x_1, \ldots, x_n\}.$$

Given $x \in \mathrm{SFP}$ with finite $X \subseteq x^\circ$, we show that $\mathbf{P}X$ is a $\mathcal{U}$-closed subset of $\mathbf{P}x$, so that the inclusion is a $\mathcal{U}$-link. Let $\mathcal{X} \subseteq \mathbf{P}X$. Then $\mathbf{P} \bigcup \{\mathcal{U}X' \mid X' \subseteq \bigcup \mathcal{X}\} \subseteq \mathbf{P}X$ is a roof of $\mathcal{X}$. For let $Y \in \mathbf{P}x$ be an upper bound of $\mathcal{X}$, and for each $y \in Y, X' \in \mathcal{X}$, let $X'_y = \{x \in X' \mid x \sqsubseteq y\}$. Then no $X'_y = \emptyset$ and every $X' = \bigcup_{y \in Y} X'_y$. Now define $\hat{Y} = \{\hat{y} \mid y \in Y\}$ where, for $y \in Y$, $\hat{y} \in \mathcal{U} \bigcup_{X' \in \mathcal{X}} X'_y$ is such that $y \sqsupseteq \hat{y} \sqsupseteq \bigcup_{X' \in \mathcal{X}} X'_y$. It is straightforward to check that $Y \sqsupseteq_{\mathrm{M}} \hat{Y} \sqsupseteq_{\mathrm{M}} \mathcal{X}$.

Next, if tuples $t \equiv_m t'(x, y)$ satisfy $\mathbf{SFP}_{\mathcal{U}}$, the induced relation is a $\mathcal{U}$-link $u : \bigcup_{j=1}^m [t_j] \to \bigcup_{j=1}^m [t'_j]$ with each $u[t_j] = [t'_j]$, $j = 1, \ldots, m$. If u extends to the isomorphism θ between the $\mathcal{U}$-closures, obviously for each $j = 1, \ldots, m$ we have

$$[t_j] \, \mathbf{P}\theta \, [t'_j]$$

So $\mathbf{C}_{\mathcal{U}}$ holds, and again $\mathbf{C}_N$ is trivial. And the (solitary) component of m-tuple t is $\bigcup_{j=1}^m |t_j|$ — if X is its $\mathcal{U}$-closure, every member of $\mathbf{P}x$ obviously has a name whose component is inside X. Hence $\mathbf{C}_*$ holds.

Finally, we show that $\mathbf{P}$ is directed. Let names s, t denote the same set. Then the extremal points in each tuple must be common, because $x \in |s| \sqsubseteq \exists y \in |t| \sqsubseteq \exists x' \in |s|$, so x maximal implies $x = y = x'$, and so on. But $[s]$ is contained in the convex-closure of its extremal points, whence $[s] = [s \cap t]$. Likewise, $[t] = [s \cap t]$.

5.4.5 Exponent, $\to$.

This has arity 2. Once again, it is obvious how to define $\to$ on finite isomorphisms.

Names.

For each $n > 0$, there is one operator Ω_n of sort $\langle 0, 1 \rangle^n$, and for each $n \geq 0$ there is one operator Ω_n^+ of sort $\langle 0, 1 \rangle^n +\!\!+ \langle 1 \rangle$. We shall write $(a_1 \to b_1, \ldots, a_n \to b_n)$ for

$\Omega_n a_1 b_1 \ldots a_n b_n$ and $(a_1 \to b_1, \ldots, a_n \to b_n, 0 \to b)$ for $\Omega_n^+ a_1 b_1 \ldots a_n b_n b$. Henceforth we shall not distinguish them because they behave uniformly with respect to this common presentation. n is the *length* of the term $(a_1 \to b_1, \ldots, a_n \to b_n)$. We now define $[\cdot]$ over *all* terms in such a way that $[t]$ is a function from x to y for $t \in \mathrm{Term}_{\langle x,y \rangle}$:

$$[(a_1 \to b_1, \ldots, a_n \to b_n)]x = \text{if } x \sqsupseteq a_1 \text{ then } b_1 \text{ else } \ldots \text{if } x \sqsupseteq a_n \text{ then } b_n \text{ else } 0$$

(just write $[a_1 \to b_1, \ldots, a_n \to b_n]$). We shall refer to the smallest i for which $x \sqsupseteq a_i$ as the i that "catches x". We shall also treat a term as a relation $\subseteq x^\circ \times y^\circ$ in the obvious way.

Then we define Name as those terms $t = (a_1 \to b_1, \ldots, a_n \to b_n)$ which satisfy $\forall i, j = 1, \ldots n$:

(N1) If $i \leq j$ and $\{a_i, a_j\} \not\vdash \{a_1, \ldots a_{i-1}\}$, then $b_i \sqsupseteq b_j$

(N2) $i \leq j$ implies $a_i \not\sqsubseteq a_j$

Note that N2 makes $[t]a_i = b_i$ for every $i = 1, \ldots, n$. Ie: $t \subseteq [t]$. The next theorem says that names denote the the proper kind of function.

5.4.5.1 THEOREM. *For name ν, $[\nu]$ is a map.*

PROOF: Let $x \sqsubseteq x'$. If $x \sqsupseteq a_i$, $x' \sqsupseteq a_i$, so if i catches x' and j catches x, then $j \geq i$. Thus $[\nu]x = b_j$, $[\nu]x' = b_i$. But then $x' \sqsupseteq \{a_i, a_j\}$, so $x' \sqsupseteq \exists a \in x^\circ \sqsupseteq \{a_i, a_j\}$. If $\{a_i, a_j\} \vdash \{a_1, \ldots, a_{i-1}\}$,then $x' \sqsupseteq a_k, \exists k < i$, contradiction. Hence $b_i \sqsupseteq b_j$.

Thus $[\nu]$ is monotonic; and since $[\nu]x = [\nu]a$ for some $a \in x|$, it is also continuous. ∎

We have two uses now of "$[\cdot]$" applied to a finite relation, viz: the function generated by a generator, and the function value of a name. The next theorem assures us that this overloading is consistent.

5.4.5.2 THEOREM. *If name ν has $\mathcal{U}$-closed left-set, so that it is a generator, its value $[\nu]$ is the same as the function $[\nu]$ it generates.*

PROOF: Write $[\nu]_{\text{val}}, [\nu]_{\text{gen}}$ to distinguish the two functions herein. Now, since $!\nu$ is finite, any $x|\nu$ has a top in $!\nu$, which must be a_i for the i that catches x. For if $j < i$ catches x, $x \sqsupseteq \exists a_k \in \mathcal{U}\{a_i, a_j\}$, and k must be $\leq j$, so k catches x. Therefore $k = j$ and $a_j \sqsupseteq a_i$, whence $a_j = a_i$, contradiction. Thus

$$[\nu]_{\text{gen}} x = \nu a_i = [\nu]_{\text{val}} x$$

∎

We now show that $\rightarrow$ is directed with respect to this naming system. To this end, we introduce a *normal form* for names, as follows.

5.4.5.3 DEFINITION. *Name $(a_1 \rightarrow b_1, \ldots, a_n \rightarrow b_n)$ is in* normal form *when it additionally satisfies*

(N3) $b_i \sqsupset b_j \ \Rightarrow\ i < j$

(N4) $i < j \ \&\ a_i \sqsupseteq a_j \ \Rightarrow\ b_i \sqsupset b_j$

We shall show that for every name ν there is a normal form $\bar{\nu} \subseteq \nu$ with $[\nu] = [\bar{\nu}]$. It then follows, if $[\nu] = [\mu]$ with

$$\bar{\nu} = (a_1 \rightarrow b_1, \ldots, a_n \rightarrow b_n)$$

and

$$\bar{\mu} = (c_1 \rightarrow d_1, \ldots, c_m \rightarrow d_m)$$

that given a_i, we have $a_i \sqsupseteq \exists c_j \sqsupseteq \exists a_{i'}$, with $b_i = d_j = b_{i'}$. So by N2, $i \leq i'$, and therefore by N4 $i = i'$. Thus $a_i = c_j$ and $a_i \rightarrow b_i$ is common to $\bar{\nu}$ and $\bar{\mu}$. So $\bar{\nu} = \bar{\mu}$. Hence $[\nu] = [\bar{\nu}] = [\mu]$ and $\bar{\nu} \subseteq \nu$ and μ — we have directedness. We now establish the normal form result.

First note that if $a \rightarrow b, a' \rightarrow b'$ are adjacent pairs of name ν with $b \not\sqsupseteq b'$ then, by N1, interchanging the pairs will not affect $[\nu]$.

5.4.5.4 LEMMA. *ν can be re-ordered to ν' with the same value and satisfying N3.*

PROOF: Induction on the length n of ν. The case $n = 1$ is trivial, so let $\nu = (a \to b) +\!\!+\, \mu$ where μ has length n, and assume that μ can be re-ordered to μ' satisfying N3. Now suppose that $b \sqsubsetneq \exists b_i$ in μ', and start moving $a \to b$ through μ' by successive interchanges. If at any stage we were to have $(\ldots, a \to b, a_{i'} \to b_{i'}, \ldots)$ with $b \sqsupseteq b_{i'}$ $(i' < i)$ then $b_{i'} \sqsubseteq b \sqsubsetneq b_i$, thereby contradicting N3 for μ'. Thus we must eventually get $a \to b$ to the immediate right of $a_i \to b_i$. Performing this operation for the largest i such that $b \sqsubsetneq b_i$ will produce a ν' satisfying N3. ∎

Now consider $\nu = (a_1 \to b_1, \ldots, a_n \to b_n)$ which satisfies N3, and let $a_i \sqsupseteq a_j$ for some $i < j$ with $b_i \not\sqsupseteq b_j$. By N1, $b_i = b_j = b$ (say), so that $a_i \to b_i$ is redundant unless $i < \exists i' < j$ with $\{a_i, a_{i'}\} \not\uparrow \{a_1, \ldots, a_{i-1}\}$ and $b_{i'} \neq b$. But in that case, N1 would give $b_{i'} \sqsubseteq b$, therefore $\sqsubsetneq b = b_j$, contradicting N3. So $a_i \to b_i$ must be redundant. Removing all such pairs modifies ν' to $\bar{\nu}$ satisfying N4. Clearly $\bar{\nu}$ still satisfies N3 and $\bar{\nu} \subseteq \nu$. And all the transformations have preserved value, so we have $[\bar{\nu}] = [\nu]$ as required.

We have still to show that the names denote precisely $(x \to y)^\circ$.

(1) $[\nu] \in (x \to y)^\circ$: Let $D \nearrow_{(x \to y)} f_0 \sqsupseteq [\nu]$, $\nu = (a_1 \to b_1, \ldots, a_n \to b_n)$. Then $f_0 a_i \sqsupseteq b_i$ implies $f a_i \sqsupseteq b_i$, $\exists f \in D$, whence $\exists f \in D$ such that *every* $f a_i \sqsupseteq b_i$. Now let x be caught by i. Then $fx \sqsupseteq f a_i \sqsupseteq b_i = [\nu]x$. So every $[\nu]$ is compact, and clearly $\neq 0$.

(2) Every $f \in (x \to y)^\circ$ has a name: Given non-zero map $f : x \to y$, let R be the set of all $r \subseteq x| \times y^\circ$ such that r is a finite generator in graph(f). Then

(a) $f = \bigsqcup_{r \in R}[r]$. For obviously every $[r]x = \bigsqcup r(x|r) \sqsubseteq fx$. But if $g \sqsupseteq \{[r] \mid r \in R\}$, let $\langle a, b \rangle \in \text{graph} f$. Then $\langle a, b \rangle \in R$, so $ga \sqsupseteq [r]a = b$, whence $\text{graph} f \subseteq \text{graph} g$ giving $f \sqsubseteq g$.

(b) Now let $r, r' \in R$. We define $s \in R$ as follows. Take $!s = \mathcal{U}^*(!r \cup !r')$. Extend r, r' to this set by setting $ra = 0$ for $a \notin !r$ and similarly for r' (which clearly does not affect their values). Now let $a \in !s$ be maximal, and suppose s defined elsewhere on $!s$. Then $\{ra, r'a\} \cup \{sa' \mid a \sqsupsetneq a' \in !s\} \sqsubseteq fa$. Choose $b \in \mathcal{U}(\mathrm{LHS})$ with $b \sqsubseteq fa$ and set $sa = b$. By induction on the size of $!s$, we obtain s with $\{[r], [r']\} \sqsubseteq s$. Hence R is directed.

(c) It remains to show that every $[r], r \in R$ has a name. Choose maximal $a \in !r$, and assume that we can now order $r \setminus \{\langle a, b\rangle\}$ to satisfy N2 (converting each $\langle a', b'\rangle$ to $a' \to b'$). Prefix $a \to b$ to this ordering, and N2 is still satisfied. And since $!r$ is $\mathcal{U}$-closed, every $\{a_i, a_j\} \vdash \{a_1, \ldots, a_{i-1}\}$, so N1 holds vacuously. This establishes (2).

In fact, we have characterised $(x \to y)^\circ$ in two ways: as the set of $[\nu]$ for name ν, and as the set of $[r]$ for generator r. Moreover, (2c) shows how to make an equivalent name out of a generator, and obviously, given name ν, $r = \{\langle a, [\nu]a\rangle \mid a \in \mathcal{U}^*(!\nu)\}$ is an equivalent generator extending ν.

We now turn to the conditions C_N and $\mathrm{C}_{\mathcal{U}}$.

C_N.

Let $x, y \in \mathrm{SFP}$. If terms $t \equiv_1 t'$ satisfy $\mathrm{SFP}_{\mathcal{U}}$, let A, A' be the $\mathcal{U}$-closures of the left-, right-set of the induced relation with $\theta : A \cong A'$ the extending isomorphism. Now N1 and N2 involve only entailments amongst the members of $!t$, and these are entirely determined within A because it is $\mathcal{U}$-closed. Thus the determination is the same for t or t', modulo θ, whence either both or neither of t, t' satisfy each of N1, N2.

$\mathrm{C}_{\mathcal{U}}$.

Again take $x, y \in \mathrm{SFP}$ and let $X \subseteq_{\mathrm{fin}} x|, Y \subseteq_{\mathrm{fin}} y^\circ$ be $\mathcal{U}$-closed. Then each map in $(X \to Y)$ is a generator, so $[\cdot] : (X \to Y) \to (x \to y)^\circ$ because each $r : X \to Y$ can (as in (2c) above) be converted into a name. Furthermore, every non-empty $x|X$, being finite directed, has a top, say $\hat{x}$. Also

$$[r] \sqsubseteq [s] \Rightarrow [r]a \sqsubseteq [s]a \quad \forall a \in X$$

$$\Rightarrow ra \sqsubseteq sa$$

$$\Rightarrow r \sqsubseteq s$$

so $[\cdot]$ is bimonotonic. Now let $R \subseteq (X \to Y)$; we show that $\{[r'] \mid r' \in \mathcal{U}R\}$ is a roof of $\{[r] \mid r \in R\}$, whereupon the right-set of $[\cdot]$ is $\mathcal{U}$-closed. Define $\hat{f} : X \to Y : a \mapsto (fa)|Y$. Then, in case $x|X \neq \emptyset$, $[\hat{f}]x = \hat{f}\hat{x} \sqsubseteq f\hat{x} \sqsubseteq fx$ (if $x|X$ is empty, $[\hat{f}]x = 0$ anyway) giving $[\hat{f}] \sqsubseteq f$. And $\widehat{[r]}a = ([r]a)|Y = ra$, so $\widehat{[r]} = r$. Then if $f \sqsupseteq \{[r] \mid r \in R\}$, $\hat{f} \sqsupseteq R$, whence $\hat{f} \sqsupseteq \exists r \in \mathcal{U}R$ implying $f \sqsupseteq [\hat{f}] \sqsupseteq [r]$ (pre-empting Chapter 6 a moment, having got the injection-pair $\langle [\cdot], \hat{\cdot} \rangle$, the doublet therein would tell us immediately that $[\cdot]$ is a $\mathcal{U}$-link).

If we now have $\nu \equiv_m \nu'(\langle x, y\rangle, \langle x', y'\rangle)$, let A, A' be the $\mathcal{U}$-closures of the zeroth component of ν, ν' and B, B' likewise of the first. Let θ_A, θ_B be the extending isomorphisms. Then for each $j = 1, ..., m$, ν_j (resp. ν'_j) is a name in $(A \to B)$ (resp. $A' \to B'$), and $[[\nu_j]_{A \to B}] = [\nu_j]$ because any $\hat{x}$ (relative to A) $\sqsupseteq a_i \in !\nu_j$ for the i that would catch x. Likewise each $\left[[\nu'_j]_{A \to B}\right] = [\nu'_j]$. But clearly each $[\nu_j]_{A \to B}$ $(\theta_A \to \theta_B)$ $[\nu'_j]_{A' \to B'}$, because $[\nu_j]_{A \to B}$ is determined entirely by the order structure on A, B.

Hence $\to$ does indeed satisfy $\mathbf{C}_{\mathcal{U}}$, and is therefore an $\mathbf{SFP}_{\mathcal{U}}$-constructor.

$\mathbf{C}_*$.

Finally, we prove $\mathbf{C}_*$. With ν, A, B as for $\mathbf{C}_{\mathcal{U}}$, consider $r : A \to B$. Now r generates $[r]$, so just as in (2c) above, r can be ordered into a name for $[r]$. Obviously the components of this name are contained in $A \cup B$, so $\mathbf{C}_*$ holds.

It is worth remarking here that in the simple case, working with $\vdash$ rather than $\mathcal{U}$, these proofs of constructor-hood become essentially those presented in [14] for information spaces.

Some of these constructors are sums or products, with respect to any of Alg or $\mathrm{Alg}_{\sqsubseteq,\vdash,\mathcal{U},*}$ because the injections involved are in all of these (notice that a join restricted to a sub lifts the original via the inclusion functor):

(1) Coalesced sum is a join which lifts disjoint union on Set, whence it is a disjunctive sum.

(2) Cartesian product (with injections as in Set_0) is an exact join but not a sum. It is not disjunctive. It lifts Cartesian product on Set, which is therefore not a sum.

(3) Coalesced product lifts that in Set_0, so the former is also a disjunctive strong product.

We conclude this chapter with a brief discussion of effectiveness.

5.5 Effectiveness

The bicontexts $\mathrm{Alg}_{\sqsubseteq,\vdash,\mathcal{U}}$ provide a natural setting for the notion of an effective, or computable, type (cf. [10]). We can say that ω-algebraic type x is $\sqsubseteq,\vdash,\mathcal{U}$-*computable* when the corresponding structure on x is decidable for finite subsets of x°. We now make this idea precise.

Let $\mathcal{P} = \bigcup_{n=1}^{\infty} \mathcal{P}_n$ be a set of *predicate-symbols*, with each $P \in \mathcal{P}_n$ of arity $\#P = n$. We consider the class of $\mathcal{P}$-*structures*, that is to say, sets x with a subset $P_x \subseteq x^{\#P}$ corresponding to every $P \in \mathcal{P}$, and construct the bicontext $\mathcal{P}$-Set = Set $\times$ $carrier_{\mathcal{P}}$ where $carrier_{\mathcal{P}}$ associates with every $\mathcal{P}$-structure its carrier.

5.5.1 Definition. *$r : x \to y \in \mathcal{P}$-Set is*

(1) $\mathcal{P}$-preserving *when for any $P \in \mathcal{P}_n$ and $x_i\ r\ y_i$, $i = 1, \ldots n$ we have*

$$\langle x_1, \ldots, x_n \rangle \in P_x \ \Rightarrow\ \langle y_1, \ldots, y_n \rangle \in P_y$$

(2) $\mathcal{P}$-matching *when r, r^- are both $\mathcal{P}$-preserving.*

5.5.2 THEOREM. *The $\mathcal{P}$-matching relations are a spanning hereditary bisub,* $\mathrm{Set}_{\mathcal{P}}$*, of* $\mathcal{P}$-Set.

PROOF: We shall show that the $\mathcal{P}$-preserving relations are a spanning hereditary sub, and the result will follow. The spanning and hereditary are both obvious, as indeed are 0-closure and closure under composition. It remains to prove D-closure. So let $D \nearrow_{\mathrm{Set}_{\mathcal{P}}[x,y]} r$, with every $d \in D$ $\mathcal{P}$-preserving, and let $p \in \mathcal{P}_n$, $x_i\ r\ y_i$ $(i = 1,...,n)$ and $\langle x_1, \ldots, x_n\rangle \in P_x$. Then $\exists d \in D$ such that $x_i\ d\ y_i$ $\forall i = 1,...,n$, whence $\langle y_1, \ldots, y_n\rangle \in P_y$. ∎

5.5.3 THEOREM. *The left- and right-sets of any* $r : x \to y \in \mathrm{Set}_{\mathcal{P}}$ *are elementary (first-order) equivalent qua sub* $\mathcal{P}$*-structures of* x *and* y.

PROOF: For any arbitrary set X, an *X-environment* is a function η from the individual variables of first-order predicate calculus to X. If η is a $!r$-environment and η' is a $r!$-environment, we write $\eta \sim \eta'$ to mean that $\eta\xi\ r\ \eta'\xi$ for every variable ξ. We shall show that, for any formula Φ and $\eta \sim \eta'$,

$$!r \models_\eta \Phi \ \Leftrightarrow\ r! \models_{\eta'} \Phi$$

where $\models_\eta$ means "satisfies using η to supply the values of the free variables".

We proceed by induction on the structure of Φ:

$\Phi = P(\xi_1, \ldots, \xi_n)$:

$$\begin{aligned}\text{LHS} &\Leftrightarrow \langle \eta\xi_1, \ldots, \eta\xi_n\rangle \in P_x \\ &\Leftrightarrow \langle \eta'\xi_1, \ldots, \eta'\xi_n\rangle \in P_y \\ &\Leftrightarrow \text{RHS}\end{aligned}$$

$\Phi = \Phi_1\ \&\ \Phi_2$:

$$\begin{aligned}\text{LHS} &\Leftrightarrow\ !r \models_\eta \Phi_1 \text{ and } !r \models_\eta \Phi_2 \\ &\Leftrightarrow r! \models_{\eta'} \Phi_1 \text{ and } r! \models_{\eta'} \Phi_2\end{aligned}$$

$$\Leftrightarrow \text{ RHS}$$

$\Phi = \neg\Psi$:

$$\begin{aligned} \text{LHS} &\Leftrightarrow\ !r \not\models_\eta \Psi \\ &\Leftrightarrow\ r! \not\models_{\eta'} \Psi \\ &\Leftrightarrow \text{ RHS} \end{aligned}$$

$\Phi = \forall\xi\Psi$:

$$\text{LHS} \Leftrightarrow\ !r \models_{\eta[\xi/a]} \Psi \quad \text{for any } a \in\ !r$$

Assume LHS and let $a\ r\ b$. Then, since $\eta[\xi/a] \sim \eta'[\xi/b]$,

$$!r \models_{\eta[\xi/a]} \Psi \Rightarrow r! \models_{\eta'[\xi/a]} \Psi$$

so $r! \vdash_{\eta'} \Phi$. And conversely gives LHS $\Leftrightarrow$ RHS. ∎

Given an m-place formula Φ with free variables $\xi_1, \ldots, \xi_m$, we shall write

$$\lambda\xi_1 \ldots \xi_m.\Phi$$

for the *derived predicate*, which denotes in $x \in \text{obj}(\mathcal{P}\text{-Set})$ the relation

$$(\lambda\xi_1 \ldots \xi_m.\Phi)_x = \{(x_1, \ldots, x_m) \mid \exists\eta\ x \models_{\eta[\xi_1/x_1,\ldots,\xi_m/x_m]} \Phi\} \subseteq x^m$$

(the particular η used in this definition is irrelevant).

We now define three particular $\mathcal{P}$s, and give the *centre* of every $x \in \text{obj}(\text{Alg})$ a $\mathcal{P}$-structure.

(1) $\mathcal{P}$ has one binary symbol Le with

$$\langle x_1, x_2\rangle \in Le_x \Leftrightarrow x_1 \sqsubseteq x_2$$

We shall denote it by "$\sqsubseteq$".

(2) $\mathcal{P}$ has an $(n+m)$-ary symbol $E^{n,m}$ for every $n, m \geq 0$, with

$$\langle x_1, \ldots, x_n, x'_1, \ldots, x'_m \rangle \in E^{n,m}_x \Leftrightarrow \{x_1, \ldots, x_n\} \vdash \{x'_1, \ldots, x'_m\}$$

We shall denote it by "$\vdash$". Note that the information spaces of [9] are simple $\vdash$-structures.

(3) $\mathcal{P}$ has an n-ary symbol $U^{A,n}$ for every finite order A with carrier $\subseteq \omega$ and $\supseteq \{1, \ldots, n\}$. Its interpretation is $\langle x_1, \ldots, x_n \rangle \in U^{A,n}_x$ iff the correspondence $\{i \mapsto x_i \mid i = 1, ..., n\}$ extends to an isomorphism $\theta : A \cong \mathcal{U}^*\{x_1, \ldots, x_n\}$. We shall denote this $\mathcal{P}$ by "$\mathcal{U}$".

If we now define, for each of these $\mathcal{P}$, a function (which we take the liberty of also calling $\mathcal{P}$)

$$\mathcal{P} : \mathrm{obj}(\mathrm{Alg}) \to \mathrm{obj}(\mathcal{P}\text{-Set}) : x \mapsto \text{the } \mathcal{P}\text{-structure of } x^\circ$$

(it does not matter whether this $\mathcal{P}$-structure is taken relative to x or x° because $\mathcal{U}^*$ of any $X \subseteq_{\mathrm{fin}} x^\circ$ is also $\subseteq x^\circ$).

Then $\mathcal{P}$ is (1,1) because the order is determined by the $\mathcal{P}$-structure in each case ($\sqsubseteq, \vdash$ are obvious, and $x_1 \sqsubseteq x_2$ iff $\langle x_1, x_2 \rangle \in U^{A,2}_x$ where $A = \{1, 2\}$ under the natural ordering).

So we can identify Alg with $\mathcal{P}$-Set $\times\, \mathcal{P}$, which in turn can be treated as a full bisub of $\mathcal{P}$-Set. Then

5.5.4 PROPOSITION. *For each of these $\mathcal{P}$, the $\mathcal{P}$-links as previously defined are just the $\mathcal{P}$-matching relations. Ie:* $\mathrm{Alg}_{\mathcal{P}} = \mathrm{Set}_{\mathcal{P}}|\mathrm{Alg}$.

We can now define object x in Alg to be $\mathcal{P}$-computable, relative to some enumeration $a : \omega \to x^\circ$, when for every $P \in \mathcal{P}_n$ the set

$$\{\langle i_1, \ldots, i_n \rangle \mid \langle a_{i_1}, \ldots, a_{i_n} \rangle \in P_x\}$$

is decidable.

Now the definition of constructor makes the $\mathcal{P}$-structure of Fx, for I-ary $\mathrm{Alg}_{\mathcal{P}}$-constructor F, determined finitely, and abstractly, by the $\mathcal{P}$-structures of the argument x_is. It is straightforward to require this determination to be computable (relative to suitable coding of names etc.), to obtain the notion of a *computable constructor* (one requirement will be that synonymity of names is decidable). Then it is immediate that a computable constructor produces computable types from computable types.

Also, the colimit of a directed diagram $[\![\cdot]\!]$ of $\mathcal{P}$-computable types is $\mathcal{P}$-computable provided the diagram itself is computable in the sense that

(1) the indexing directed set, N, has an enumeration, n, with respect to which its ordering is decidable, ie: $\{\langle i, j\rangle \mid n_i \leq n_j\}$ is decidable, and

(2) if the centre of $[\![n_i]\!]$ is enumerated by a^i, then

$$\{\langle i, i', j, j'\rangle \mid {a^i}_{i'} \, [\![n_i, n_j]\!] \, {a^j}_{j'}\}$$

is also decidable.

Actually, by considering all the derived operators, we can make the limit functor F^ω a constructor when F is. Then for the canonical diagram generated by F, condition (1) is immediate and condition (2) is simply that synonymity of names over the colimit be decidable. In other words, $(\cdot)^\omega$ will preserve computability.

Finally here, we note that although (the centres of) simple types are finitely axiomatisable in first-order logic (by the order axioms plus one stating dualmost-completeness), SFP types are not axiomatisable at all.

We show SFP is not closed under ultrapowers, whence it is not closed under elementary equivalence, so that none of SFP, its complement or SFP $\setminus$ Simp are axiomatic (see [15] for the relevant results).

So let $\mathcal{F}$ be an ultrafilter on ω containing the cofinite sets. For $n \in \omega$, let $A_n{}^\circ = \{a, b\} \cup \{0, \ldots, n\}$ with $\{a, b\} \sqsubseteq i$, $\forall i \in \{0, \ldots, n\}$ (see Diagram 5.4). Let A be the coalesced sum of the A_n (write a_n for the copy of a in A_n, etc.). Then A is obviously SFP. Let Φ be the formula

$$\lambda a, b, x.(a \sqsubseteq x \ \& \ b \sqsubseteq x \ \& \ \forall y(a \sqsubseteq y \ \& \ b \sqsubseteq y \ \& \ y \sqsubseteq x \Rightarrow y = x))$$

$\Phi(a, b, x)$ says that x is a minimal upper bound of $\{a, b\}$. Now consider $B \stackrel{\text{def}}{=} A^\omega / \mathcal{F}$ and let $a, b, n \in B$ $(n \in \omega)$ be the equivalence-classes of the tuples $\langle a_i \rangle_{i=0}^\infty, \langle b_i \rangle_{i=0}^\infty$ and $\langle a_i \rangle_{i=0}^{n-1} + \!\!\! + \langle n_i \rangle_{i=n}^\infty$. Then $A \models \Phi(a_i, b_i, n_i)$ for every $n \in \omega$, $i \geq n$. Thus $\{i \mid A \models \Phi(a_i, b_i, n_i)\} \in \mathcal{F}$, whence $B \models \Phi(a, b, n)$. Also, distinct m, n yield distinct sequences in B, so $\{a, b\}$ has no roof in B. It follows that B is not SFP.

Chapter 6

The Doublet SFP2i, SFP_*

In this chapter we shall construct a doublet between a bisub of the map-pair bicontext **Type2** as "high", and the Set-based bicontext SFP_* discussed in Chapter 5, as "low". We shall first consider them individually, then define the doublet.

6.1 Alg2i

6.1.1 Definition. Alg2i = Int(**Type2**)|Alg. SFP2i *and* Simp2i *likewise.*

Convention.

As remarked in 3.3, given $f = \langle \acute{f}, \grave{f} \rangle \in$ Alg2i, we shall use f as a map to mean $\acute{f}$ where no confusion will result.

We now consider the strong members of Alg2i. Unfortunately, these do not constitute a sub, not being closed under composition as the following example shows.

Let $x = \{0, 1, 2\}$ under their natural ordering and define $f, g : x \to x$ by

$$\acute{f} = \grave{f} = \{0, 1 \mapsto 0,\ 2 \mapsto 2\}$$

$$\acute{g} = \grave{g} = \{0 \mapsto 0,\ 1, 2 \mapsto 1\}$$

(see Diagram 6.1). They are both strong, but their composite

$$\langle \{0, 1 \mapsto 0,\ 2 \mapsto 1\},\ \lambda x.0 \rangle$$

is not. However, a strong interior does determine its reverse uniquely.

6.1.2 THEOREM. *If f, g are strong with $\acute{f} = \acute{g}$, then $\grave{f} = \grave{g}$.*

PROOF: Let f, g satisfy the premiss. Then

$$
\begin{aligned}
\grave{f} &= \grave{(}f \mathbin{;} f^{-} \mathbin{;} f) \\
&= \grave{f} \mathbin{;} \acute{f} \mathbin{;} \grave{f} \\
&= \grave{f} \mathbin{;} \acute{g} \mathbin{;} \grave{f} \\
&= \grave{f} \mathbin{;} \acute{(}g \mathbin{;} g^{-} \mathbin{;} g) \mathbin{;} \grave{f} \\
&= \grave{f} \mathbin{;} \acute{g} \mathbin{;} \grave{g} \mathbin{;} \acute{g} \mathbin{;} \grave{f} \\
&= \grave{f} \mathbin{;} \cancel{\acute{f} \mathbin{;} \grave{g} \mathbin{;}} \acute{f} \mathbin{;} \grave{f} \\
&\sqsubseteq \grave{g}
\end{aligned}
$$

Likewise, $\grave{g} \sqsubseteq \grave{f}$, whence the result. ∎

6.1.3 DEFINITION. *For $f : x \to y \in \mathrm{Alg2i}$, $\mathrm{Fix}f = \{\langle x, y\rangle \mid y = \acute{f}x \ \&\ x = \grave{f}y\}$.*

It is then immediate that

6.1.4 PROPOSITION.

(1) Fix$f$ *is (1,1)* $\subseteq f$

(2) $\mathrm{Fix}(f^{-}) = (\mathrm{Fix}f^{-})$

By part (2), we can unambiguously write Fixf^{-}. We shall call $f : x \to y \in \mathrm{Alg2i}$ *algebraic* when Fix$f$ is algebraic (relative to $x \times y$). It is immediate that f is algebraic iff f^{-} is.

6.1.5 LEMMA. *If f is strong with $f^R \sqsubseteq g^L$ (equivalently $\grave{f} \mathbin{;} \acute{f} \sqsubseteq \acute{g} \mathbin{;} \grave{g}$), then $f \mathbin{;} g$ is strong.*

PROOF: Since $f \mathbin{;} f^{-} \mathbin{;} f \sqsubseteq f$ holds for *any* $f \in \mathrm{Alg2i}$, we must show the reverse inequality for $f ; g$. Now f strong implies $f = f ; f^R$, so

$$\begin{aligned} f ; g ; g^- ; f^- ; f ; g &= f ; g^L ; f^R ; g \\ &\sqsupseteq f ; f^R ; f^R ; g \\ &= f ; f^R ; g \\ &= f ; g \end{aligned}$$

∎

6.1.6 COROLLARY. *If strong f, g fit, $f ; g$ is strong.*

6.1.7 DEFINITION. *For $f : x \to y \in \mathrm{Alg2i}$, the* centre *of f is*

$$f^\circ \stackrel{\text{def}}{=} \mathrm{Fix} f \cap (x^\circ \times y \cup x \times y^\circ)$$

6.1.8 LEMMA.

(1) $1^\circ = 1$

(2) $(f^-)^\circ = (f^\circ)^-$

(3) *f° is a $\sqsubseteq$-link (ie: in $\mathrm{Alg}_{\sqsubseteq}$).*

(4) *$f^\circ \subseteq x^\circ \times y^\circ$, whence $f^\circ = (\mathrm{Fix} f)^\circ$.*

(5) *If $f : x \to y,\ g : y \to z$, then $(f ; g)^\circ = f^\circ ; g^\circ$*

(6) *If $f \in \mathrm{SFP2i}$, f° is a generator.*

PROOF: (1),(2) and (3) are obvious.

(4) Let $\langle a, b\rangle \in f^\circ$ with $a \in x^\circ$. Let directed D pass b. Then $f^- D$ passes, and therefore dominates, a. So D dominates $f(f^- D)$ which dominates b.

(5) Let $\langle a, c\rangle \in (f ; g)^\circ$ and let $b = fa$. Then

$$\begin{aligned} a &\sqsupseteq f^-(fa) \\ &\sqsupseteq f^-(g^-(g(fa))) \\ &= a \end{aligned}$$

so $a = f^- b$. But $gb = c$, so

$$
\begin{aligned}
b &\sqsupseteq g^-(gb) \\
&= g^- c \\
&\sqsupseteq f(f^-(g^- c)) \\
&= fa \\
&= b
\end{aligned}
$$

giving $b = g^- c$. Thus $\langle a, b\rangle \in f^\circ$ and $\langle b, c\rangle \in g^\circ$, whence $\langle a, c\rangle \in f^\circ \,;\, g^\circ$. The converse is trivial.

(6) Obviously f° is monotonic. We must show that its left-set is $\mathcal{U}$-closed. Let $X \subseteq_{\text{fin}} !(f^\circ)$ and let $a \in \mathcal{U}X$. Since $a \sqsupseteq X$, $f^-(fa) \sqsupseteq X$, so $f^-(fa) \sqsupseteq \exists a' \in \mathcal{U}X$, whence $a \sqsupseteq f^-(fa) \sqsupseteq a'$. But this implies $a = a'$, so $a \in !(f^\circ)$. ∎

We shall abbreviate $x|!(f^\circ)$ to $x|f$. From parts (1),(2),(3) and (6) we obtain

6.1.9 COROLLARY. *For $f \in \mathbf{SFP2i}$, $f^\circ \in \mathbf{SFP}_*$.*

Henceforth in this section, we confine our workings to **SFP2i**.

6.1.10 THEOREM. *$(\cdot)^\circ$ is an exact functor from* **SFP2i** *to* $\mathbf{SFP}_*$.

PROOF: It remains only to prove continuity. First, let $f \sqsubseteq g$, with $\langle a, b\rangle \in f^\circ$. Then $ga \sqsupseteq b$, whence

$$
\begin{aligned}
a &\sqsupseteq g^-(ga) \\
&\sqsupseteq g^- b \\
&\sqsupseteq f^- b \\
&= a
\end{aligned}
$$

So $g^- b = a$, and conversely. Thus $(\cdot)^\circ$ is monotonic. Now let $D \nearrow f$ with $\langle a, b\rangle \in f^\circ$. Then $fa = \bigsqcup_{d\in D} da = b$, so $da = b$, $\exists d \in D$. Likewise $d'b = a$, $\exists d' \in D$. Thus $\langle a, b\rangle \in \bigcup_{d\in D} d^\circ$. ∎

6.2 SFP$_*$

Since any $u : x \to y$ in Alg$_*$ is a generator, we can extend it to the map $[u] : x \to y$ (recall that $[u]x = \bigsqcup u(x|u)$). This extension has the following additional property.

6.2.1 LEMMA. *If $u : x \to y$, $v : y \to z$, then $[u\,;v] \sqsubseteq [u]\,;[v]$, with $=$ if u, v fit.*

PROOF:

$$[u\,;v]x = \bigsqcup v\ u(x|(u\,;v))$$

$$(1) \qquad = \bigsqcup_{a \in x|(u;v)} v(ua)$$

and

$$[v]([u]x) = \bigsqcup_{a \in x|u} [v]a$$

$$(2) \qquad = \bigsqcup_{a \in x|u}\ \bigsqcup_{b \in (ua)|v} vb$$

Now if $a \in x|(u\,;v)$, then $ua \in !v$ and $a \in x|u$, therefore $ua \in (ua)|v$. So (1) $\sqsubseteq$ (2).

But if $u^R \subseteq v^L$, $ua' \in (ua')|v\ \forall a' \in x|u$, and $!(u\,;v) = !u$, so the inner sup of (2) $= v(ua)$, whence the equality.

And if $u^R \supseteq v^L$, then $!(u\,;v) = u^-(!v)$, so

$$a \in x|u \ \&\ b \in (ua)|v \ \Rightarrow\ u^- b \in x|(u\,;v)$$

$$\Rightarrow\ vb = v\ u\ u^- b$$

Thus (2) $\sqsubseteq$ (1), and again we get equality. ∎

6.2.2 COROLLARY. *The pair $\langle [u], [u^-] \rangle$ is a strong Alg2i.*

PROOF: u, u^- fit, so the Lemma gives $[u]\,;[u^-] = [u\,;u^-] \sqsubseteq 1$ and $[u]\,;[u^-]\,;[u] = [u\,;u^-\,;u] = [u]$. Similarly the other way round. ∎

Convention.

In line with our equivocation between f and $\acute{f}$, we shall refer to the pair $\langle [u], [u^-] \rangle$ simply as $[u]$.

6.2.3 THEOREM. $[\cdot]$ *is a linear thread functor from* **SFP**$_*$ *to* **SFP2i**.

PROOF: If u, v fit we can apply Lemma (6.2.1) to either u, v or v, u. It remains only to prove linearity (and a fortiori, continuity). First monotonicity. Let $u \subseteq v$. Then $x|u \subseteq x|v$ giving $[u]x \sqsubseteq [v]x$. Now let $u = \bigcup V$. Then $!u = \bigcup_{v \in V} !v$, $x|u = \bigcup_{v \in V}(x|v)$ and $u(x|u) = \bigcup_{v \in V} v(x|v)$, whence $[u]x = \bigsqcup_{v \in V}[v]x$. ∎

6.3 THE DOUBLET

We now relate $(\cdot)^\circ$ and $[\cdot]$ together to construct the doublet.

6.3.1 THEOREM. $(\cdot)^\circ : \mathbf{SFP2i} \leftrightarrows \mathbf{SFP}_* : [\cdot]$ *is a doublet (with* **SFP2i** *high).*

PROOF: The functors are certainly thread functors. Obviously $0^\circ = \emptyset$. In addition we have

(a) If $f : x \to y \in \mathbf{SFP2i}$, $x|f^\circ \sqsubseteq x$, so $f^\circ(x|f^\circ) \sqsubseteq fx$. Thus $[f^\circ] \sqsubseteq f$.

(b) Let $u : x \to y \in \mathbf{SFP}_*$, with $\langle a, b\rangle \in u$. Then $a|u = a$ and $u(a|u) = b$, so $[u]a = b$. Likewise $[u]^- b = a$. Thus $\langle a, b\rangle \in [u]^\circ$, yielding $u \subseteq [u]^\circ$. ∎

We shall see shortly that this doublet gives a match between the various functors and limit constructions in **SFP** and the usual ones in **SFP2i**. Now, though, we look at the stable arrows and at the special case of **Simp**.

6.3.2 THEOREM. $[f^\circ] = f$ *iff* f *is strong algebraic.*

PROOF:

($\Rightarrow$) We show that every $[u]$ is algebraic; we already know that it is strong. Let $\langle x, y\rangle \in \mathrm{Fix}[u]$. Then $x = [u\,;u^-]x = \bigsqcup(x|u)$. Likewise $y = \bigsqcup(y|u^-)$. But obviously $\langle x, y\rangle|u = (x|u)\,;u\,;(y|u^-)$, so $\bigsqcup\langle x, y\rangle|u = \langle x, y\rangle$.

($\Leftarrow$) Assume f is strong algebraic. Then

$$\begin{aligned}
[f^\circ]x &= \bigsqcup f^\circ(x|f^\circ) \\
&= \bigsqcup f(x|f) \\
&= f \bigsqcup (x|f). \\
&= f(f^-(fx)) \quad \text{because } f \text{ is strong algebraic} \\
&= fx
\end{aligned}$$

∎

Thus we have $[f^\circ] \sqsubseteq f$ with equality if f is either an injection or a projection.

Not every strong f is algebraic as the following example shows.

Take $x = \{[0,0]\} \cup \{[0,r), [0,r] \mid 0 < r \leq 1\} \subseteq \mathbb{Q}$ (the rational numbers) under $\subseteq$. Let us write $r^=$ for $[0,r]$ and $r^<$ for $[0,r)$. Then x is simple with centre $\{r^= \mid r > 0\}$. Define $f : x \to x$ to have the graph $\{\langle r^=, s^= \rangle \mid 0 \leq s < r\} \cup \{\langle 0^=, 0^= \rangle\}$. Then f is strong (with itself as reverse) but $f^\circ = \emptyset$, which generates the pair $\langle \lambda x.0^=, \lambda x.0^= \rangle$, not $\langle f, f \rangle$. In fact, $\mathrm{Fix} f = x \setminus x^\circ$.

However, we do have

6.3.3 THEOREM. *If f is strong with f, f^- both central, f is algebraic.*

PROOF: Let $\langle x, y \rangle \in \mathrm{Fix} f$. Then $\langle x, y \rangle = \langle f^- fx, fx \rangle$. Let $S = \{\langle f^- fa, fa \rangle \mid 0 \neq a \in x|\}$. The conditions on f imply that $S \subseteq f^\circ$. Thus $\bigsqcup S = \langle f^- fx, fx \rangle = \langle x, y \rangle$. ∎

This theorem cannot be reversed, though. For let $x = \omega + 2$ under the natural ordering, so that $x^\circ = x \setminus \{0, \omega\}$. Let $f : x \to x$ be

$$\{n \mapsto n \mid n < \omega\} \cup \{\omega \mapsto \omega\} \cup \{\omega + 1 \mapsto \omega\}$$

Then f is a strong interior (again self-reverse) with $f^\circ = \{\langle n, n \rangle \mid n < \omega\}$ and $\mathrm{Fix} f = f^\circ \cup \{\langle \omega, \omega \rangle\}$, so it is algebraic. But obviously f is not central.

6.3.4 THEOREM. *Every $u \in \mathbf{SFP}_*$ is stable.*

PROOF: We must show that $[u]^\circ \subseteq u$. Let $\langle a, b\rangle \in [u]^\circ$. Then $a = [u\,;u^-]a = \bigsqcup(a|u)$, whence $a \in (a|u)$, ie: $a \in {!u}$. Likewise $b \in u!$. Then $b = [u]a = \bigsqcup u(u|a) = ua$. ∎

Simple.

If $u : x \to y \in \mathbf{Simp}_\vdash$, we can define $[u]$ without requiring $\mathcal{U}$-closure (which in this case is dual-closure) of the ends. For if $X \subseteq_{\text{fin}} x|u$, then X is consistent, ie: $X \not\vdash \emptyset$. Thus $uX \not\vdash \emptyset$. so every finite subset of $u(x|u)$ has a sup, and hence $[u]x$ exists as the sup of the (directed) dual-closure of $u(x|u)$. The doublet construction is still valid with this extended $[\cdot]$, so we obtain the doublet

$$(\cdot)^\circ : \mathbf{Simp2i} \leftrightarrows \mathbf{Simp}_\vdash : [\cdot]$$

And of course, the stable arrows at the low level are just $\mathbf{Simp}_*$.

6.4 COLIMITS

Every injection, i, in $\mathbf{SFP2i}$ is stable, for it is obviously strong, and algebraic because $!(\mathrm{Fix}\,i) = 1$. So to every colimit in either of $\mathbf{SFP2i}$ or $\mathbf{SFP}_*$, there is a matching one in the other; in particular, every injective diagram in $\mathbf{SFP2i}$ has a colimit, which is the standard limit construction normally used to solve type equations. We show below that the constructions defined in Chapter 5 to represent the usual type transformations match the standard functors on $\mathbf{SFP2i}$. Thus all the solution limits match across the doublet.

It is a property of SFP types (see [5]) that they can be obtained as colimits in $\mathbf{SFP2i}$ of diagrams with finite nodes. We can translate this property down to $\mathbf{SFP}_*$, and it does in fact give a better handle on the centre of an SFP type. The property also means that we can use the functor extension machinery of (3.7) to extend any functor defined on some $K \cap \mathbf{Fin}$, with $\mathbf{SFP}_* \subseteq K \triangleleft^= \mathbf{SFP}$, to K

itself. This parallels the manner in which the **SFP2i**-functors introduced in [5] are defined.

6.5 Matching Functors

Here we show that the constructors defined in Chapter 5, which were all within $\mathbf{SFP}_*$, match their usual counterparts in **SFP2i**. Also those defined on $\mathbf{Simp}_\vdash$ match relative to the fuller doublet.

Having already shown that the constructors are stable, by (3.10) it suffices to show (using the notation of that section) that for any $u \in \mathbf{SFP}_*^I$ ($\mathbf{Simp}_*^I$) — arity I —

(MF1) $$[F_L u] = F_H[u]$$

(this actually says that F_H lifts F_L via $[\cdot]$)

and that for those F_L defined on $\mathbf{Simp}_\vdash^I$, for any u therein

(MF2) $$[f_L u] \sqsubseteq F_H[u]$$

Sum.

Let $u : x \to x',\ v : y \to y'$ be in either $\mathbf{SFP}_*$ or $\mathbf{Simp}_\vdash$. Then

$$x|(u+v) = x|u$$

$$y|(u+v) = y|v$$

so that

$$[u+v]x = [u]x$$

$$[u+v]y = [v]y$$

This actually also follows from the facts that $[\cdot]$ is linear and that $+$ is a sum at both levels with matching injections. So

$$[u+v] = \left[(left^- \,;\, u \,;\, left) \cup (right^- \,;\, v \,;\, right)\right]$$

where $left$ and $right$ are the injections of the sum

$$= \left([left]^{-} ; [u] ; [left]\right) \sqcup \left([right]^{-} ; [v] ; [right]\right)$$

(all the necessary fits hold)

$$= [u] + [v]$$

Hence $+_H, +_L$ are in fact matching sums.

Product.

Let u, v be as for $+$. Then $\langle x, y\rangle|(u \times v) = (x|u) \times (y|v)$, so

$$[u \times v] = [u] \times [v]$$

Notice that in both cases, the equality holds everywhere in the low level.

Power.

Let $u : x \to y \in \mathbf{SFP}_*$. We shall compare the graphs of $\mathrm{P}[u]$ and $[\mathrm{P}u]$. First, if A, B are compact in $\mathrm{P}x$, with each $b \in B$ being a directed sup, say $b = \bigsqcup B_b$, then it is easy to see that $A \sqsubseteq_{\mathrm{M}} B$ iff $A \sqsubseteq_{\mathrm{M}}$ some section of $\{B_b \mid b \in B\}$, where by a *section* we mean some $B' \subseteq_{\mathrm{fin}} \bigcup_{b \in B} B_b$ such that each $b \in B$, $B' \cap B_b \neq \emptyset$.

Now let A, B be compact in $\mathrm{P}x, \mathrm{P}y$ respectively. Then $(\mathrm{P}[u])A$ is the convex closure of the Cantor closure of $[u]A$, whence

$$B \sqsubseteq_{\mathrm{M}} (\mathrm{P}[u])A \;\Leftrightarrow\; B \sqsubseteq_{\mathrm{M}} [u]A$$

(*) $$\Leftrightarrow\; B \sqsubseteq_{\mathrm{M}} \text{ some section of } \{u(a|u) \mid a \in A\}$$

and $A|\mathrm{P}u$ is the set of sections of $\{a|u \mid a \in A\}$, so

$$B \sqsubseteq_{\mathrm{M}} [\mathrm{P}u]A \;\Leftrightarrow\; B \sqsubseteq_{\mathrm{M}} u(\text{some section of } \{a|u \mid a \in A\})$$

(**) $$\Leftrightarrow\; B \sqsubseteq_{\mathrm{M}} \text{ some section of } \{u(a|u) \mid a \in A\}$$

(*) and (**) are the same statement.

Exponent.

Again, let $u : x \to x'$, $v : y \to y'$ be in either $\mathbf{SFP}_*$ or $\mathbf{Simp}_\vdash$. Recall from 5.4.5 that for $f \in (x \to y)$, $f| = \{[\mu] \mid \text{name } \mu \subseteq \text{graph} f\}$. Then

$$f|(u \to v) = \{[\mu] \in (f|) \mid !\mu \subseteq !u \ \& \ \mu! \subseteq !v\}$$

Moreover, in the $\mathbf{SFP}_*$ case since $!u, !v$ are $\mathcal{U}$-closed, we can always extend μ to a generator.

We shall compare graphs again, this time of $[u \to v]$ and $[u] \to [v]$ by dint of some "compact point chasing".

Let a, b be compact in x', y' respectively, and let $f \in (x \to y)$. Then

$$b \sqsubseteq [u \to v] f a \ \Leftrightarrow \ b \sqsubseteq (u \to v)[\mu] a = [(u \to v)\mu] a$$

for some name $\mu \subseteq \text{graph} f$ with $!\mu \subseteq !u, \mu! \subseteq !v$

$$(\dagger) \qquad \Leftrightarrow \ b \sqsubseteq v(\mu a') \quad \exists a' \in !\mu, \ ua' \sqsubseteq a$$

$$\Rightarrow \ b \sqsubseteq [v] \ f \ [u^-] a$$

So the inequality holds in all cases. Restrict now to $\mathbf{SFP}_*$. Then

$$b \sqsubseteq [v] \ f \ [u^-] a \ \Leftrightarrow \ b \sqsubseteq [v] b_1, \quad \exists b_1 \sqsubseteq f a_1, \quad \exists a_1 \sqsubseteq [u^-] a$$

(all a's and b's compact). But

$$a_1 \sqsubseteq [u^-] a \ \Leftrightarrow \ a_1 \sqsubseteq \exists a_2 \in !u, \ u a_2 \sqsubseteq a$$

So the RHS becomes $b \sqsubseteq [v] b_1$, $\exists b_1 \sqsubseteq f a_2$ with $u a_2 \sqsubseteq a$. But

$$b_1 \sqsubseteq f a_2 \ \Leftrightarrow \ b_1 \sqsubseteq [\mu] a_2$$

for some generator $\mu \subseteq \text{graph} f$ with $!\mu \subseteq !u, \mu! \subseteq !v$

$$\Leftrightarrow \ b_1 \sqsubseteq \mu a_3, \quad \exists a_3 \in a_2 | \mu$$

But $a_2 | \mu$ has a top, say $\hat{a}_2$, whence

$$b_1 \sqsubseteq f a_2 \ \Leftrightarrow \ b_1 \sqsubseteq \mu \hat{a}_2, \quad \exists a_2, \ u a_2 \sqsubseteq a$$

So RHS now becomes $b \sqsubseteq [v](\mu \hat{a}_2)$, $\exists a_2$ with $u a_2 \sqsubseteq a$. And

$$b \sqsubseteq [v](\mu a_2) \ \Leftrightarrow \ b \sqsubseteq v b_2, \quad \exists b_2 \in (\mu \hat{a}_2) | v$$

(‡) $$\Leftrightarrow b \sqsubseteq v(\mu\hat{a}_2) \quad \text{because } \mu\hat{a}_2 \in {!v}$$

Now, given (†), take $a_2 = a'$ whence $\hat{a}_2 = a'$, to get (‡). Conversely, take $a' = \hat{a}_2$; then, since $ua' \sqsubseteq ua_2 \sqsubseteq a$, we get (†).

We conclude this chapter by pointing out the property, perhaps slightly jarring, of the doublet presented here that $[\cdot]$ does not preserve regularity. All the low-level arrows are regular interiors (so $(\cdot)^\circ$ preserves regularity trivially), but not all the stable high-level arrows are regular, as a straightforward counter-example shows.

Let $x^\circ = \{0, 1, 2\}$, $y^\circ = \{0, 1\}$, both under their natural ordering. Define

$$u = \{0 \mapsto 0, 1 \mapsto 1\}$$

$$f = \{0, 1 \mapsto 0, 2 \mapsto 1\}$$

Then $[u] = \{0 \mapsto 0,\ 1, 2 \mapsto 1\}$ whence $f \leq [u]$, but $[u^L]\,; f < f$.

Chapter 7

Applications and Examples

We begin by presenting the induction principle associated with a functor, which was one of the aims stated in Chapter 1.

With the notation of (3.9), let $p \in \mathrm{P}x$ be a solution of K-functor F in medium $f : x \to Fx$. Then p is a fixed-point of $\binom{f}{f}$, so we have

7.1 PROPOSITION. *$\binom{f}{f}$-seed a generates p (via f) iff $\binom{f}{f}$ is an induction on $[a, p]$.*

PROOF:

($\Rightarrow$) If $\binom{f}{f}q \leq q$ and $a \leq q \leq p$ then $p \leq q$

($\Leftarrow$) If $\binom{f}{f}(f /_a f) = f /_a f$ then $p = f /_a f$ ∎

This means that $\binom{f}{f}$ provides induction proofs that a property on x, already known to contain a, is the whole of p. Thus $\binom{f}{f}$ is a kind of "relative" induction principle. In particular, if p is the inductive solution, $\binom{f}{f}$ is a "proper" induction principle on p.

When F is a (monadic) constructor on the Set-like bicontext C, the principle $\binom{f}{f}_F$ on x is

$$p \mapsto \{[\nu] \mid \nu \in \mathrm{Name} \ \& \ |\nu| \subseteq p\}$$

We now look at the particular constructors of Chapter 5. It must be emphasised that these inductions yield proofs of universality *relative to the centre* of the inductive solution. To obtain a proof that *all* of x has a given property, the property must be $D, 0$-closed to catch 0 and the limit points.

With the exception of exponent, which requires a little work, all these inductions are immediately intuitively appealing and can be stated (at the expense of a slight weakening by not restricting the statement to compact points) without reference to names at all. Here are some examples ($X = FX$ means $x = F^{\omega}0$).

Examples.

$\underline{(\cdot)}$: Let $\Omega = \underline{\Omega}$. Then $\Omega \cong \omega + 1$ under its natural ordering and the induction principle is MI because, if $\Omega' \subseteq \Omega$,

$$\underline{\Omega}' = \{0\} \cup \{n+1 \mid n \in \Omega'\}$$

And D-closure catches ω itself.

$+$: Let $N = \{0\}_{\perp} + N$ Then $N \cong \omega_{\perp}$, and the induction principle is MI together with "$\perp$ has the property" because, for $N' \subseteq N$,

$$1 + N' = \{0\} \cup \{n+1 \mid n \in N'\}$$

$\times$: Let $List = \{nil\}_{\perp} + \omega \times List$ This yields standard list structural induction with

$$List'(\subseteq List) \mapsto \{nil\} \cup \omega_{\perp} \times List'$$

P: Let $X = \mathbb{T} + \mathbf{P}X$ where $\mathbb{T} = \{tt, ff\}_{\perp}$. The induction principle can be weakened slightly to the convenient:

To prove $X' = X$ (for $X' \subseteq X$), prove $\mathbb{T} \subseteq X'$ and that $x \in X'$ when $x \subseteq X'$.

$\rightarrow$: In the case of exponent, the induction principle prima facie depends on the names. Let us consider the example of

$$E_{\infty} = \mathbb{T} + (E_{\infty} \rightarrow E_{\infty})$$

To prove that $X \subseteq E_{\infty}$ is the whole set requires proving that (a) $\mathbb{T} \subseteq X$ and (b) $(a_1 \rightarrow b_1, \ldots, a_n \rightarrow b_n) \in X$ if $\{a_i, b_i \mid i = 1, \ldots, n\} \subseteq X$.

But if we can establish (b) with the weakened assumption $\{b_i \mid i = 1, ..., n\} \subseteq X$, we have removed name dependency, and explicit mention of compactness, because $\{b_1, \ldots, b_n\}$ is just the range of $a_1 \to b_1, \ldots, a_n \to b_n$.

Thus we have the convenient (albeit weaker) induction

$$X = E_\infty \quad \text{when} \quad \mathrm{T} \subseteq X \ \& \ \forall f \in (E_\infty \to E_\infty)(fE_\infty \subseteq X \Rightarrow f \in X)$$

This is, in fact, exactly the transformation we mentioned in Chapter 1 for obtaining a structural induction on E_∞. But the difference is that there it was necessary, whereas here it is only cosmetic. The transformation is actually an application of a more general one whereby the diagonal of a functor is replaced by a specialisation of it as follows.

Let $F : C^I \to C$ be a K-functor ($K \triangleleft^= C$) and let strong $p \leq x$ be a solution of F_Δ in the medium $f : x \to Fx$. Then p is also a solution of any $F_{\bar{p},i}$ ($i \in I$), where $\bar{p} = \langle p \rangle_{i' \in I}$.

If $p' \leq p$, then

$$\begin{aligned}
\binom{f}{f}_{F_\Delta} p' &= f \,;\, F_\Delta p' \,;\, f^- \\
&= f \,;\, F\bar{p}' \,;\, f^- \\
&\leq f \,;\, F(p[i/p']) \,;\, f^- \\
&= \binom{f}{f}_{F_{\bar{p},i}} p'
\end{aligned}$$

So, qua induction on p, $\binom{f}{f}_{F_\Delta} \succeq \binom{f}{f}_{F_{\bar{p},i}}$

We now present a more extended example, showing some sample induction proofs.

Let I, J be countable sets, and let I^* be the set of strings over I under the prefix ordering (write $i = (is)_0$, $s = (is)_1$ and put $0_0 = 0_1 = 0$, where 0 is the null string). Then $I^+ = I^* \setminus \{0\}$ is dualmost-complete and therefore the centre of a simple type $\tilde{I}$ of "streams" over I. Likewise $J^*, J^+, \tilde{J}$.

Now define the type *Proc* as the inductive solution of the functor

$$F : \mathrm{Simp}_{\mathcal{U}} \to \mathrm{Simp}_{\mathcal{U}} : f \mapsto (f.J)^I$$

where $(\,).J$ means a J-fold sum. Thus *Proc* represents "processes" which take an input in I, deliver an output in J and renew with another process. For $p \in Proc$ write $p?$ for the unique $j \in J$ such that $p \in Proc_j$, and conversely $p.j$ for p qua element of $Proc_j$ (let $p.0 = 0 \in Proc.J$). Also define $Trans = \tilde{I} \to \tilde{J}$.

Now I^+ possesses the ordinary structural induction principle

$$P \mapsto \{is \in I^+ \mid s \in P \cup \{0\}\}$$

and similarly J. And the functor F gives the induction principle on *Proc*

$$P \mapsto \{p \in Proc \mid \forall i \in I,\ p_i \in P\}$$

We shall now illustrate the use of these inductions. Define functions

$tail : I \to Trans \to Trans$ by $tail\ i\ f\ s = (f\ is)_1$

$ext : Proc \to Trans : p \mapsto$ the canonical extension of $\lambda(is \in I^+).(p_i?)\ ext\ p_i\ s$

$int : Trans \to Proc : f \mapsto \langle int(tail\ i\ f).(fi)_0\rangle_{i \in I}$

Both ext, int are strict, and $tail\ i\ 0 = 0$ for any $i \in I$.

7.2 THEOREM. $\langle ext, int\rangle : Proc \rightleftarrows Trans$ *in* **Simp2i**.

PROOF: Let $P = \{p \in Proc \mid int(ext\ p) = p\}$. It is obviously D,0-closed. Given $p \in Proc$, assume that $p_i \in P\ \forall i \in I$. Then for any $i \in I$,

$$\begin{aligned} int(ext\ p)_i &= int(tail\ i\ (ext\ p)).(ext\ p\ i)_0 \\ &= int(ext\ p_i).(p_i?) \\ &= p_i.(p_i?) \\ &= p_i \end{aligned}$$

Thus $ext\,;int = Proc$.

For the opposite direction, lacking an induction on $Trans$, we use induction on $\tilde{I}$ as follows. Let $P' = \{s \in \tilde{I} \mid \forall f \in Trans,\ ext(int\ f)s \sqsubseteq fs\}$. Again this property is obviously D,0-closed. Assume $s \in P'$. Then

$$\begin{aligned}
ext(int\ f)\ is &= ((int\ f)_i?)\ ext(int\ f)_i s \\
&= (fi)_0\ (ext(int(tail\ i\ f))s) \\
&\sqsubseteq (fi)_0\ (tail\ i\ f\ s) \\
&= (fi)_0(f\ is)_1 \\
&\sqsubseteq f\ is
\end{aligned}$$

∎

Now define

$inc : Trans \to Trans : f \mapsto$ the canonical extension of $\lambda(is).(fi)_0\ inc(tail\ i\ f)s$

(inc is therefore strict).

7.3 THEOREM. *$\langle inc, inc\rangle$ is a part of $Trans$ (relative to* Simp2i*).*

PROOF: Here again we use induction on $\tilde{I}$. We want $inc \sqsubseteq Trans$ so take

$$P = \{s \in \tilde{I} \mid \forall f \in Trans,\ inc\ f\ s \sqsubseteq fs\}$$

Clearly it is D,0-closed. Then, assuming $s \in P$,

$$\begin{aligned}
inc\ f\ is &= (fi)_0\ inc(tail\ i\ f)s \\
&\sqsubseteq (fi)_0\ tail\ i\ f\ s \\
&= (fi)_0\ (f\ is)_1 \\
&\sqsubseteq f\ is
\end{aligned}$$

∎

If we "lower" inc across the doublet of Chapter 6, we obtain a property of functions in $Trans^{\circ}$. Compact f has this property when for any $is \in \tilde{I}$

(1) $(f\ is)_0 = (fi)_0$

(2) $(f\ is)_1$ also has the property.

Such functions are "incremental", hence the name *inc*. We conclude this section by showing that the incremental *Trans*'s are isomorphic to *Proc*.

7.4 THEOREM. $ext\,; inc = ext$.

PROOF: Induction on *Proc* again; take $P = \{p \in Proc \mid inc(ext\ p) = ext\ p\}$. Again it is clearly D,0-closed. Given $p \in Proc$, assume that for any $i \in I$, $p_i \in P$. Then

$$\begin{aligned} inc(ext\ p)\ is &= (ext\ p\ i)_0\ inc(tail\ i(ext\ p))s \\ &= (p_i?)\ inc(\lambda s'.(ext\ p\ is')_1)s \\ &= (p_i?)\ inc(ext\ p_i))s \\ &= (p_i?)\ (ext\ p_i\ s) \\ &= ext\ p\ is \end{aligned}$$

∎

7.5 COROLLARY. $\langle ext, int\rangle : Proc \cong\rangle \langle inc, inc\rangle$.

7.6 THEOREM. $inc \sqsubseteq int\,; ext$.

PROOF: Induction on $\tilde{I}$ this time. Let $P = \{s \in \tilde{I} \mid \forall f \in Trans,\ inc\ f\ s \sqsubseteq ext(int\ f)s\}$. Again D,0-closure is immediate. Assume $s \in P$. Then

$$\begin{aligned} inc\ f\ is &= (fi)_0\ inc(tail\ i\ f)s \\ &\sqsubseteq (fi)_0\ ext(int(tail\ i\ f))s \\ &= (int\ f_i?)\ ext(int\ f)_i s \\ &= ext(int\ f)\ is \end{aligned}$$

∎

7.7 COROLLARY. $\langle ext, int\rangle : Proc \cong \langle inc, inc\rangle$.

It follows that $\langle inc, inc\rangle$ is stable in the doublet, ie: strong algebraic.

We finish off with a small example showing how two parallel Type arrows, say $f, g : x \to y$, where y has an induction, may be proven comparable (equal) by means of that induction. For $f \sqsubseteq g$ iff the subset

$$\{y \mid \forall x,\ y \sqsubseteq fx \ \Rightarrow\ y \sqsubseteq gx\}$$

is equal to y, which statement is amenable to induction on y. Similarly, equality can be obtained by changing the $\Rightarrow$ to $\Leftrightarrow$. When fixed-point induction is used for such equality proofs, it requires two separate inequality proofs; the method here compresses these into a single proof.

Chapter 8

Conclusion

We have introduced the idea of a bicontext and shown how it provides a common framework for both limit constructs and structural induction. We conclude by considering some defects, loose ends, connections, extensions and applications that indicate possible further developments of the work presented.

Functors.

The basic definition of a functor appears only as a vehicle for obtaining the special cases that are of real interest. Since these cases are rather loosely related, their intersection is very weak. There is, however, a more uniform expression of these cases, which, although not providing any more substantial commonality, makes functor into a parametric rather than *ad hoc* "polyconcept". The underlying theme in this uniformity is the preservation of squares whose top and bottom belong to some particular subs K, K' respectively (call them K, K'-squares and functors). A functor may also be K, K'-exact when it preserves exact K, K'-squares.

Upper and lower functors (on C) can then be obtained as Obj,C- and C,Obj-functors respectively. The limit theory requires only an Adj,Adj-functor, with the additional assumption of being upper for the limit functor to work. And the doublet theory goes through with a lower Adj,Adj-functor that is Iso,Adj-exact; a thread functor between interior bicontexts has these properties, so the doublet of Chapter 6 survives.

Other examples of functors fit nicely into this mould; the exponentiation functor $A \rightarrow ?$ on the bicontext of types and arbitrary relations is an upper Map,Map-functor which is Map,Obj-exact.

Sums.

In 3.1 it was observed that there is an exact analogy between the zero objects of a bicontext and those of an *Ab*-category. It was also noted at the beginning of 3.5 that the definition of (binary) sum in a bicontext is formally that of biproduct in an *Ab*-category, although in the latter the operation is an Abelian group and not a sup. Unfortunately, the similarity is superficial, because the sup operation in a bicontext is partial, and composition does not generally respect it anyway. It is therefore not necessarily true that a sum is a simultaneous product and coproduct, although it will hold for a complete linear bicontext (such as **Set** with disjoint union). Nor will a sum generally be constructible from a product or coproduct in an underlying context. For example, Cartesian product is a product on $\mathbb{T}\mathbf{ype}$ and a sum on $\mathbb{T}\mathbf{ype2}$, but it is not a coproduct on $\mathbb{T}\mathbf{ype}$. In a typical coproduct diagram there may be many or no candidates for the requisite factorising arrow.

Limits.

The definition of limit has the same "sup of right-ends" form as a sum, except that it is not required to be a functor. It may be better to include such a requirement in the definition, making limit (relative to a particular net) an exact analogue of sum.

Links.

It is unsatisfactory simply to be unable to find naming systems that make exponentiation or powertype $\mathbf{SFP}_{\llcorner}$-constructors — proofs are needed. There is also the possibility of finding other subs of $\mathbf{Alg}_{\llcorner}$ for which $\rightarrow$ or P are constructors (partial, if the subs in question are not closed under $\rightarrow$ or P).

Bicontexts.

Much of the general theory of contexts and bicontexts has been left unstudied. All the paths that have been opened up but not followed offer ways of developing and extending the present work.

Induction.

Given a $D, 0$-closed property p on a solution x of constructor F, one attempts to prove that $p = x$ via an induction proof that $p^\circ (\stackrel{\text{def}}{=} p \cap x^\circ) = x^\circ$. This requires (abusing notation) $Fp^\circ \leq p^\circ$, which is equivalent to $Fp^\circ \leq p$ since $Fp^\circ \leq x^\circ$. Thus we actually have a stronger induction principle than if we applied F directly to p, even were this possible. But the price is that we do not have induction principles for non-algebraic subsets of a type. We have not pursued this question, nor have we broached proof-theoretic questions concerning the embodiment of induction principles in formal rules of inference. Moreover, the relationships between the kinds of induction herein and those used in other areas of mathematics and logic warrant further study.

Types.

The work herein is concerned with the mathematical foundations of denotational semantics and program proving, and as such is about constructing data-types and establishing induction principles thereon. An important area of current research is polymorphism in programming languages, which involves theories not only of types, but also of *sub-types*. The description of subsets of a type as loops on that type (in an appropriate bicontext, such as the various "links") would seem relevant here, and even restricted to algebraic subsets is sufficient to include the ideal model of [17]. The latter also makes use of metric properties; this line of investigation may yield useful bicontexts of metric spaces.

The question of the effective presentation of a type was touched on only briefly in 5.5. This is another important problem area, and the suggestions made there need to be developed.

Teaching.

In [9], Scott developed the information systems approach to the theory of simple types mainly for the didactic purpose of making it more palatable than it was previously. Such an exercise often improves both understanding and presentation of a topic. The original motivation for the work herein was similar, particularly with regard to the "set-based" treatment of algebraic types, and it is hoped that the effect, for the more general theory, is also similar.

References

[1] Gunter, C., *Profinite solutions for recursive domain equations*, Technical Report CMU-CS-85-107, Carnegie-Mellon University (1985).

[2] Markowsky, G. and Rosen, B., *Bases for chain-continuous posets*, IBM Journal of Research and Development **20** (1976), 138–147.

[3] Niño-Salcedo, J., *On Continuous Posets and Their Applications*, Doctoral Dissertation, Tulane University (1981).

[4] Paulson, L., *Deriving structural induction in LCF*, Lecture Notes in Computer Science **173** (1984), 197–214.

[5] Plotkin, G. D., *A powerdomain construction*, SIAM Journal of Computing **5** (1976), 452–487.

[6] Scott, D. S., *Outline of a mathematical theory of computation*, Proc. 4th Ann. Princeton Conf. on Information Sciences and Systems (1970), 169–176.

[7] ———, *Continuous lattices*, Lecture Notes in Mathematics **274** (1972), 97–136.

[8] ———, *Data types as lattices*, SIAM Journal of Computing **5** (1976), 522–587.

[9] ———, *Lectures on a mathematical theory of computation*, Technical Report, no. PRG-19, Oxford University Computing Laboratory (1981).

[10] Smyth, M. B., *Effectively given domains*, Theoretical Computer Science **5** (1977), 257–274

[11] ———, *The largest cartesian-closed category of domains*, Theoretical Computer Science **27** (1983), 109–119.

[12] Smyth, M. B. and Plotkin, G. D., *The category-theoretic solution of recursive domain equations*, SIAM Journal of Computing **11** (1982), 761–783.

[13] Wand, M., *Fixed-point constructions in order-enriched categories*, Theoretical Computer Science **8** (1979), 13–30.

[14] Winskel, G. and Larsen, K., *Using information spaces to solve recursive domain equations effectively*, Lecture Notes in Computer Science **173** (1984), 109–130.

[15] Bell, J. L. and Slomson, A. B., "Models and Ultraproducts", North-Holland, 1969.

[16] MacLane, S., "Categories for the Working Mathematician", Springer-Verlag, 1971.

[17] MacQueen, D., Plotkin, G. D. and Sethi, R., *An ideal model for recursive polymorphic types*, Proc. 11th Ann. ACM Symp. on Principles of Programming Languages (1984), 165–174.

Appendix

Diagrams

This Appendix contains all the diagrams referred to in the text, numbered according to their chapter.

Diagram 2.1

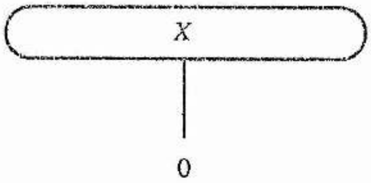

Diagram 2.2

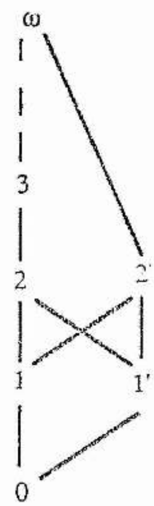

Diagram 2.3

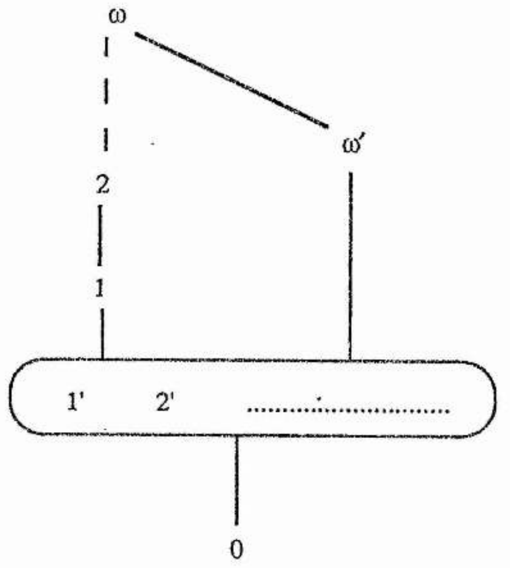

Diagram 2.4

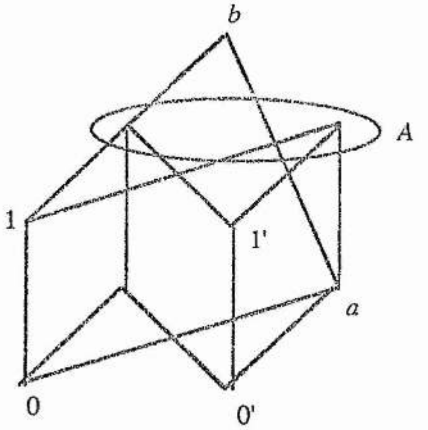

Diagram 3.1

$$\begin{tikzcd}
x \arrow[r, "top"] \arrow[d, "left"'] & y \arrow[d, "right"] \\
x' \arrow[r, "bottom"'] & y'
\end{tikzcd}$$

Diagram 3.2

$$\begin{tikzcd}
\bullet \arrow[r, "top"] \arrow[d, "left"'] & \bullet \arrow[r, "top'"] \arrow[d, "right = left'"] & \bullet \arrow[d, "right'"] \\
\bullet \arrow[r, "bottom"'] & \bullet \arrow[r, "bottom'"'] & \bullet
\end{tikzcd}$$

Diagram 3.3

$$\begin{tikzcd}
Fi \arrow[r, "Ff"] \arrow[d, "t_i"'] & Fj \arrow[d, "t_j"] \\
Gi \arrow[r, "Gf"'] & Gj
\end{tikzcd}$$

Diagram 3.4

$$\begin{tikzcd} x_i \arrow[r, "\eta_{x,i}"] \arrow[d, "f_i"'] & Fx \arrow[d, "Ff, g"] \\ y_i \arrow[r, "\eta_{y,i}"'] & Fy \end{tikzcd}$$

Diagram 3.5

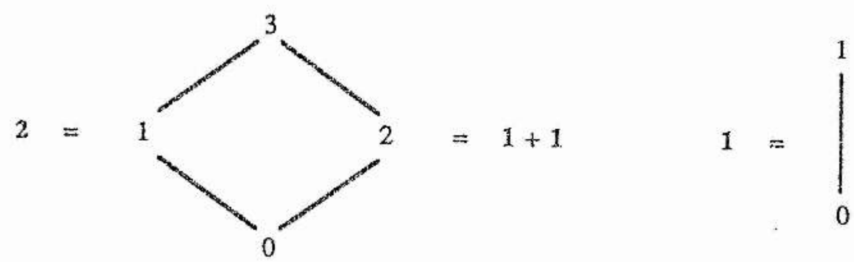

Diagram 3.6

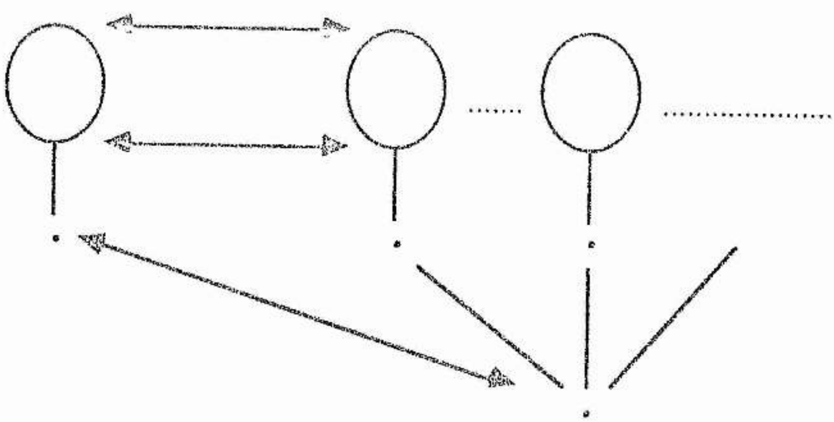

Diagram 3.7

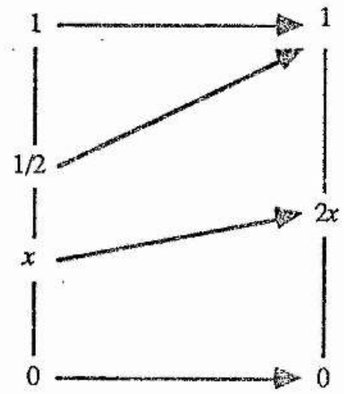

Diagram 3.8

$$\begin{tikzcd}
{[\,]} \arrow[r, "c"] \arrow[d, "r"'] & c_\infty \arrow[d, "r_\infty"] \\
{[\,]} \arrow[r, "d"'] & d_\infty
\end{tikzcd}$$

Diagram 3.9

$$\begin{tikzcd}
x \arrow[r, "f"] \arrow[d, "r"'] & Fx \\
y \arrow[r, "g"] \arrow[d, "s"'] & Fy \\
z \arrow[r, "h"] & Fz
\end{tikzcd}$$

Diagram 4.1

$p \otimes q =$

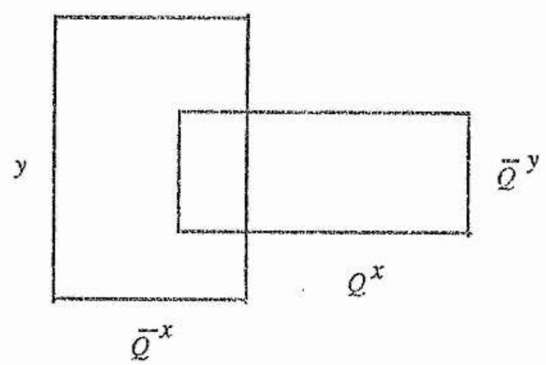

Diagram 5.1

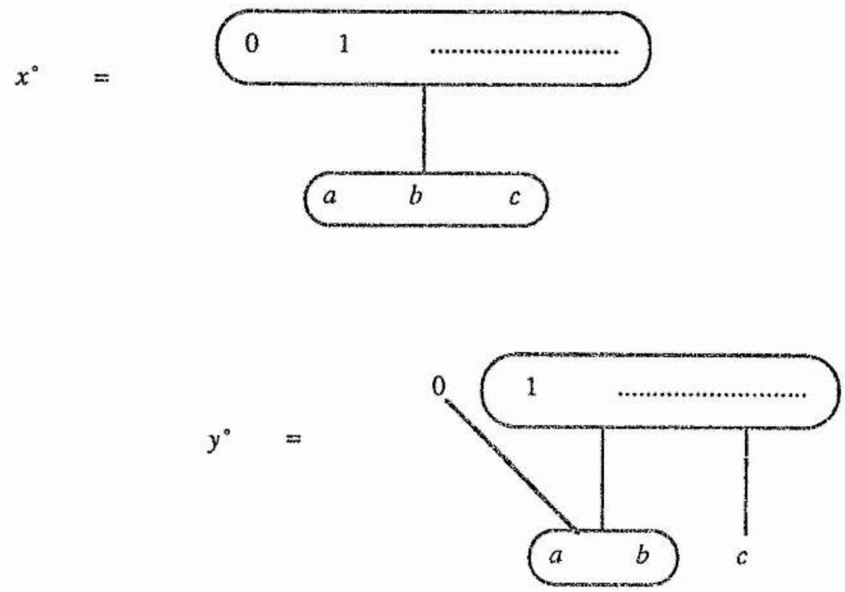

Diagram 5.2

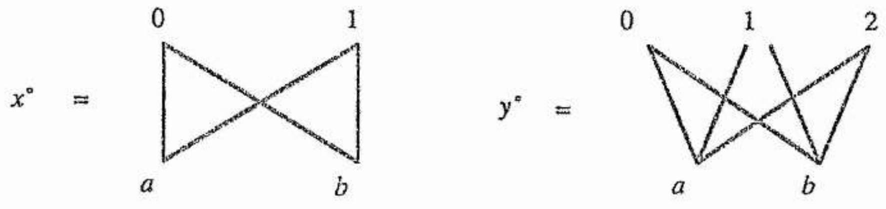

Diagram 5.3

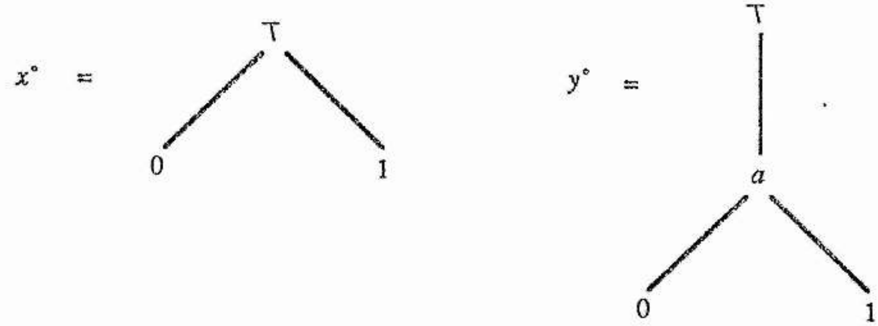

Diagram 5.4

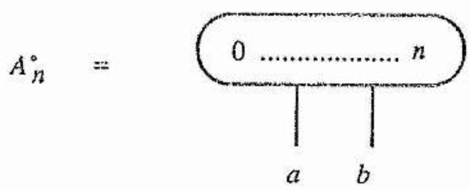

Diagram 6.1